Technical Report 908

The Coupled Depth / Slope Approach to Surface Reconstruction

John G. Harris

MIT Artificial Intelligence Laboratory

AI-TR 908 The Coupled Depth/Slope Surface Reconstruction John G. Harris

This blank page was inserted to preserve pagination.

THE COUPLED DEPTH/SLOPE APPROACH TO SURFACE RECONSTRUCTION

by

John G. Harris

Revised version of a thesis submitted to the Department of Electrical Engineering and Computer Science on May 16, 1986 in partial fulfillment of the requirements for the Degree of Master of Science.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Army contract number DACA76-85-C-0010 and in part by DARPA under Office of Naval Research contract N00014-85-K-0124.

Abstract

Reconstructing a surface from sparse sensory data is a well known problem in computer vision. Early vision modules typically supply sparse depth, orientation and discontinuity information. The surface reconstruction module incorporates these sparse and possibly conflicting measurements of a surface into a consistent, dense depth map. The coupled depth/slope model developed here provides a novel computational solution to the surface reconstruction problem. This method explicitly computes dense slope representations as well as dense depth representations. This marked change from previous surface reconstruction algorithms allows a natural integration of orientation constraints into the surface description, a feature not easily incorporated into earlier algorithms. In addition, the coupled depth/slope model generalizes to allow for varying amounts of smoothness at different locations on the surface.

This computational model helps conceptualize the problem and leads to two possible implementations – analog and digital. The model can be implemented as an electrical or biological analog network since the only computations required at each locally connected node are averages, additions and subtractions. A parallel digital algorithm can be derived by using finite difference approximations. The resulting system of coupled equations can be solved iteratively on a mesh-of-processors computer, such as the Connection Machine. Furthermore, concurrent multi-grid methods are designed to speed the convergence of this digital algorithm.

Thesis Supervisor: Dr. Thomas F. Knight

Title: Professor of Electrical Engineering

This empty page was substituted for a blank page in the original document.

Acknowledgments

Thanks to:

Demetri Terzopoulos for careful explanations of his surface reconstruction work,

Tom Knight for his willingness to suggest and entertain unusual ideas,

Christof Koch, Tommy Poggio, Allessandro Verri, Alan Yuille, Jim Little, and the rest of the vision people in the lab who are always ready to listen to new approaches,

Manfred Trummer for many conversations and for helping find eigenvalues with the EISPACK routines,

Nick Trefethen for suggesting how to extend local Fourier analysis to a coupled system of equations,

Harry Voorhees for the use of many of his vision utilities,

Lance Glasser for his support and circuit expertise,

Scott Wills for help on many things,

Paul Penfield for strong guidance during my MIT career,

the rest of the NIAL group - Trenchard More and IBM, Carl McCrosky, Mike Jenkins, Mike McIlrath and Jean Michel - for supporting me in my first two years of graduate school and for keeping in touch thereafter,

Eve Sullivan for helping proofread this report,

Anita Flynn for her inspiration, for careful readings of many rough drafts, and for being my number one friend throughout everything,

the rest of my friends who put up with me through best and worst times,

my family for their constant love and support.

This empty page was substituted for a blank page in the original document.

Contents

This empty page was substituted for a blank page in the original document.

CHAPTER 1
INTRODUCTION

This report studies surface reconstruction, a well known problem in computer vision. Computer vision is an active topic of current research for two reasons. First, we want to build artificial vision systems for use in robotics, factory automation, parts inspection and other industrial applications. Second, we study computer vision in order to understand the information processing tasks necessary for biological vision systems.

This computational approach to vision was first proposed by Marr [1982]. Marr partitioned visual processing into three levels: early vision, in which various modules detect shape, depth or discontinuities from raw images; intermediate vision, which uses the outputs from early vision modules and reconstructs surfaces in a viewer-centered coordinate frame, and late vision, in which processes such as recognition are performed on objects in an object-centered coordinate frame.

In the years since Marr's original work, a tremendous amount of research has been done in early vision. Underlying much of this work is the concept that early vision algorithms exhibit extraordinarily fine-grained locality and thus fit neatly onto the coming wave of massively parallel computers. It has not been clear, however, how the speedups achievable in early vision algorithms, through the use of parallel computation, can be extended to later levels of visual processing.

This report concentrates on increasing the processing speed for intermediate vision by developing a novel way of conceptualizing surface reconstruction. This new approach, called the coupled depth/slope model, lends itself to implementation on a fine-grained parallel machine. Furthermore, it also maps naturally to

an analog network realization for an even faster solution. Analog network ideas are not new to computer vision. Horn [1974] designed an analog network for the solution of the lightness from brightness problem. Poggio and Koch [1985] used regularization analysis to devise analog hardware to solve most of the problems in early vision. Here we develop an analog network model to solve the full surface reconstruction problem in intermediate vision.

Perhaps an even more interesting result of this work is the emerging resemblance of the artificial vision systems we are constructing to the biological vision systems that we are studying. This convergence of the two research directions, although not required, is not coincidental either.

Necessary criteria for biologically feasible information processing systems have been compiled by [Ullman 1979, p. 88; Grimson 1981, p. 163; Knight 1983, p. 7] and can be summarized as follows:

1. Massive parallelism - a large number of independent nodes performing computations.

2. Uniform, simple nodes.

3. Local interconnect scheme between nodes.

4. Fault tolerance - performance does not depend critically on any one node.

These same constraints for biological feasibility carry over to the domains of VLSI design and the construction and programming of massively parallel architectures. In VLSI design, parallelism on chip provides more efficient use of real estate; uniform, simple cells shorten layout time; local interconnect reduces the need for long wires, and fault tolerance increases yield, as a chip can still function even though some parts are broken. In parallel architecture design and use, highly parallel algorithms keep the maximum number of processors busy; uniformity in both hardware and software shortens the time for design, testing

and debugging; local communication in algorithms allows for simpler communication hardware, and fault tolerance enables the machine to keep running even though some processors or connections may become disabled. These same constraints that hamper VLSI and computer architecture designers seem to exist in the biological world as well. Since these three implementation mediums are bound by similar constraints, it should not be surprising that similar algorithms result.

Parallelism, locality, simplicity and uniformity are features of the algorithms developed in this report. This work concentrates on the artificial machine vision approach to computer vision. We try to reconstruct the original surface as fast and as accurately as possible in both digital and analog hardware. Some psychophysical tests have shown that there exist some similarities between the coupled depth/slope approach and human vision. These similarities are welcome and noted, but are not required.

Through surface reconstruction we seek to find a plausible surface given sparse, noisy and possibly conflicting measurements of an original surface [Grimson 1981; Terzopoulos 1984]. Surface interpolation and surface approximation are two special cases of surface reconstruction. A surface interpolation algorithm fits a smooth surface over sparse but exact depth measurements. Surface approximation extends this idea to fit surfaces over noisy depth measurements. The final computed surface is not required to pass through the data measurements. Surface reconstruction is a more general problem, combining information from many modalities. This report considers the integration of depth and orientation constraints with surface discontinuity information. A solution of this problem is extremely computationally intensive. Chapter 2 describes surface reconstruction more fully.

Chapter 3 develops the new coupled depth/slope model for surface recon-

struction. This network model is shown to be well suited for the surface reconstruction problem. The coupled depth/slope model is an improvement on the thin plate model used by Terzopoulos [1984] and other biharmonic models for the following reasons:

1. Orientation constraints are integrated into the surface solution very naturally. Other models, such as the thin plate model, contain no inherent notion of slope.

2. The model generalizes to arbitrary levels of smoothness. In fact it directly implements the controlled continuity models of Terzopoulos [1986].

3. Both digital and analog network implementations of the coupled depth/slope model are possible and both are discussed in this report. The parallel version of the standard biharmonic mask used by Terzopoulos and Grimson does not converge; an extra local weighting step must be added to regain convergence. The parallel version of the coupled depth/slope algorithm described here, converges by simply alternating between depth and slope calculations.

4. Both the analog and digital implementations of this model require *only* nearest neighbor communication throughout the entire algorithm. This communication feature, plus the fact that the nodes are performing extremely simple arithmetic and averaging operations, simplifies both the analog and the parallel digital implementations.

Chapter 4 derives the finite difference equations and the resulting iterative scheme necessary for a digital algorithm. This algorithm is well suited for a fine-grain massively parallel computer, such as the Connection Machine [Hillis 1985]. The single-grid digital implementation of this model suffers from slow convergence rate problems common to all local iterative methods. We propose a simple but fast extension to analog network theory to speed up convergence

through the use of coarser levels of resolution. This idea is inspired by the concurrent multigrid work of Kuo [1985] and Terzopoulos [1984], but is a much simpler realization. This multigrid network is useful in speeding up both the analog and digital realizations of the coupled depth/slope model.

Ideas for an analog implementation of the coupled slope/depth network are detailed in Chapter 5. Analog networks are interesting for a number of reasons. Most important for this work is that these networks are extremely fast. For example, they can compute solutions to the surface reconstruction problem several orders of magnitude faster than the corresponding parallel digital algorithms. Also analog networks are interesting because the algorithms they implement may be similar to the ones used by biological systems for vision. This raises the possibility for someday "growing" and "programming" neurons to create special purpose vision systems. Chapter 6 discusses promising research directions based upon this work.

CHAPTER 2
BACKGROUND

Surface reconstruction is an integral part of the approach to vision described by Marr [1982]. Marr proposed several intermediate stages in his representational framework for deriving shape information from images. His representational stages are as follows:

- Greyscale images - arrays of intensity information presented as the input to the visual system.
- Edges - the primal sketch makes explicit the sharp changes of intensity in images.
- $2\frac{1}{2}$-D sketch - an explicit representation of the depth and orientation at each point on the visible surfaces, plus discontinuities in its smoothness. The $2\frac{1}{2}$-D sketch is taken from the same viewer centered coordinate system where the original images were sampled.
- The 3-D model representation - an object centered coordinate system that represents explicitly the volumes of space that objects occupy.

Early vision modules include the stereo module, the "shape from" modules, and the edge detection module, which provide sparse and possibly conflicting sets of data for input to the surface reconstruction module. For example, stereo [Marr and Poggio 1979] supplies an extremely sparse set of depth values, the shape from shading module [Horn 1975] provides local surface orientations, and edge detection [Hildreth 1980; Canny 1983] provides possible locations for surface discontinuities. The relationship of the surface reconstruction module to these earlier stages is diagrammed in Figure 2-1.

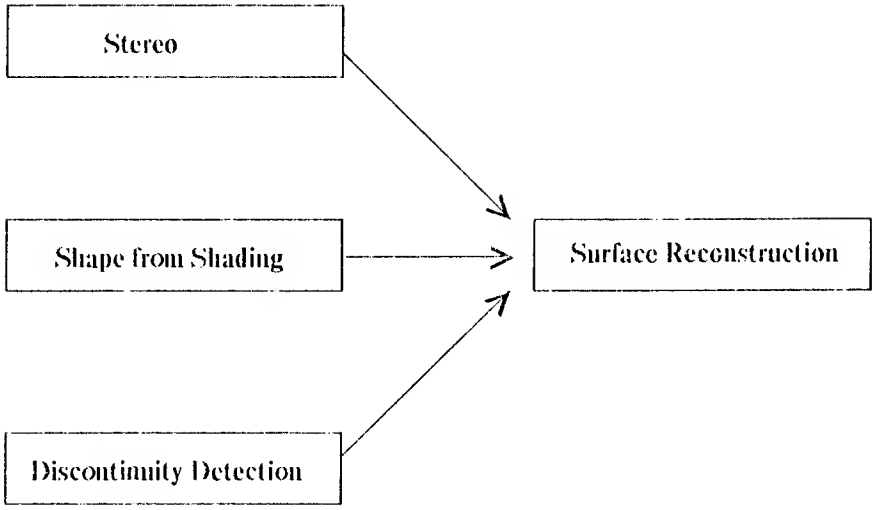


Figure 2-1. Simplified view of early and intermediate vision.

In this report, these early vision stages are considered as black boxes. We do not look inside them to see how they work; we only process their outputs with the surface reconstruction module. We do, however, open up the surface reconstruction black box and explain its operation. In the future, we will want to blur the clean distinction between early and intermediate vision to improve the overall system. For example, the stereo module and the surface reconstruction module would both produce more accurate results if there were feedback between the two systems. There is also evidence that the early vision modules interact with one another [Poggio 1984]. The discontinuity detector would operate much better if it had a closer interface to the stereo module, to the surface reconstruction, and to other visual cues such as structure from motion. Nonetheless, placing a clear dividing line between early vision modules and surface reconstruction suffices for this report.

Grimson [1981] implemented the Marr-Poggio stereo algorithm. This algorithm calculates a sparse set of depth values from the disparities of all the

matched zero crossings in stereo images. Grimson was faced with the problem of interpolating a dense surface from a sparse set of depth measurements. The problem is that there are an infinite number of surfaces which pass through the measured points. Which of these surfaces is most likely to be the original surface from which the original images were taken?

Grimson reasoned that there should be no wild undulations in the surface in between measured depth points. If wild undulations were present, then it would be likely that more points would be marked by zero crossings and produce depth measurements in the stereo module. He called this the "no news in good news" constraint. Because there is no unique solution, surface interpolation is an ill-posed problem in the sense of Hadamard, and therefore must be "regularized" [Poggio and Torre 1984]. The "no news is good news constraint" leads to the necessary stabilizing functional which ensures the uniqueness of the solution.

Using this smoothness constraint of Grimson's, we present a very simple treatment of the mathematics for the two-dimensional surface interpolation case. Surface interpolation requires finding a surface which exactly passes through sparsely measured depth points. These measurements are exact and no other types of measurements are included in the final interpolated surface. A possible set of such measurements are depicted in Figure 2-2. We want to find a surface which passes exactly through these points.

Another way of stating the "no news is good news constraint" is that we would like to minimize the first derivative between the data points. Given sparse depth points u in the x-y plane, we would like to minimize the following energy functional:

$$E_1 = \int\int (\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2 \, dx \, dy \tag{2.1}$$

The energy E_1 can be thought of as the energy of a rubber membrane [Courant and Hilbert 1953, p. 247]. This functional should only be minimized in the

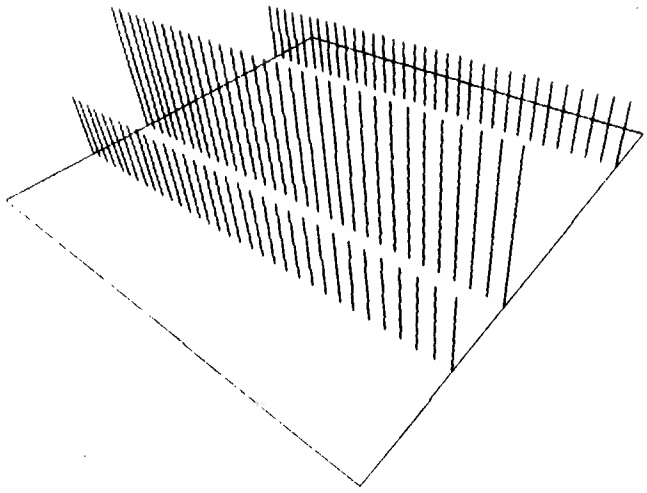


Figure 2-2. Sparse 2D depth points

regions between the depth points. The exact values of the solution surface at the depth points are fixed.

We now use the Euler-Lagrange equations [Courant and Hilbert 1962] to find an expression for the stationary point where the minimum energy must occur. An application of the Euler Lagrange equations to (2.1) yields the following solution:

$$\nabla^2 u = 0 \tag{2.2}$$

in between the constraint points (where $\nabla^2 = \left(\frac{\partial}{\partial x}\right)^2 + \left(\frac{\partial}{\partial y}\right)^2$). This is equivalent to fitting a flexible membrane between the constraint points. The membrane solution to the example problem is shown in Figure 2-3.

The membrane solution can also be modeled with a mesh of resistors. Here, the voltages are proportional to depths in the viewed surface. The energy term, equation (2.1) corresponds to the energy dissipated in all of the resistors in the mesh. Equation (2.2) is a restatement of Kirchoff's current law that all the currents flowing into a node must sum to zero. Constraints are added by applying voltage sources sparsely throughout the mesh.

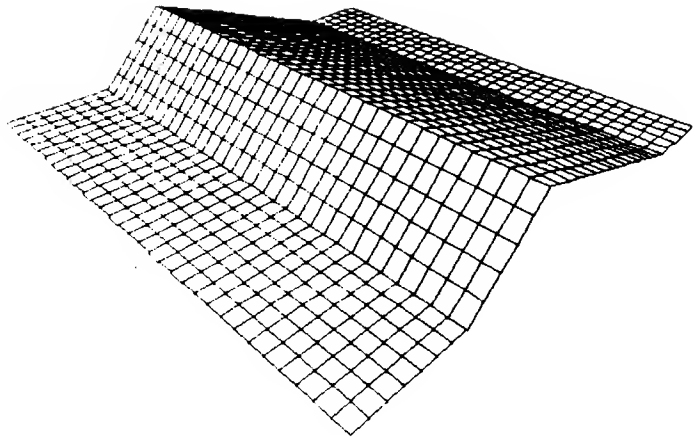


Figure 2-3. Membrane Solution.

There are three fundamental problems with the membrane solution of surface reconstruction:

1. As seen in Figure 2-3, the membrane does not extrapolate beyond the constraint points. The surface is always flat outside the given points.
2. A membrane does not have enough rigidity to incorporate orientation constraints.
3. It has been shown with random dot stereograms that humans do not interpolate surfaces by fitting straight lines in this way [Grimson 1981, p. 102]. We tend to fit smoother surfaces that do not bend sharply at the data points , for example see Figure 2-4.

A more realistic interpolation operator must be derived. A better operator results from an energy functional which minimizes the sharp bends in the solution, minimizing the change in the first derivative. This means minimizing the second derivative, so

$$E_2 = \int\int (\frac{\partial^2 u}{\partial x^2})^2 + (\frac{\partial^2 u}{\partial y^2})^2 \, dx \, dy \tag{2.3}$$

Grimson calls this energy the quadratic variation of the surface. It can also be modeled as the energy of a thin plate [Courant and Hilbert 1953]. Applying

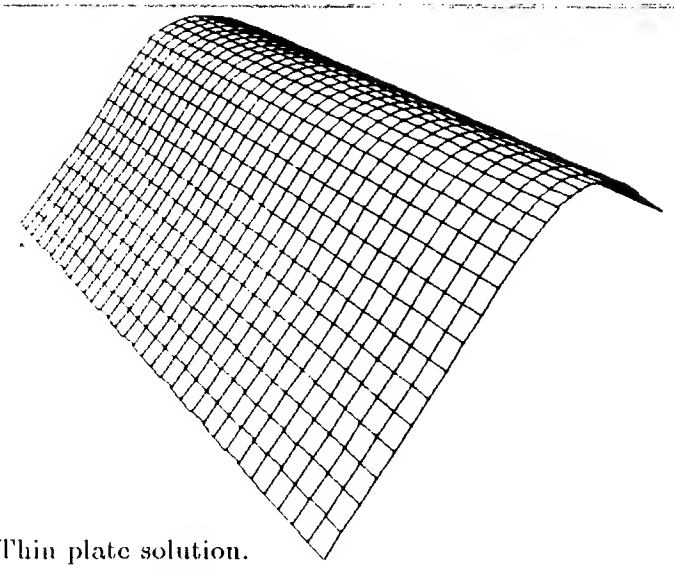


Figure 2-4. Thin plate solution.

Euler's equations to minimize E_2, we get

$$\nabla^2\nabla^2 u = 0 \tag{2.4}$$

which is equivalent to fitting cubic splines in each interval. Solution of this equation yields the curve surface shown in Figure 2-4. Notice that extrapolation is done around the exterior and the surface changes smoothly.

Grimson developed an iterative algorithm based upon minimizing the quadratic variation E_2. He worked on the surface approximation problem since he did not deal with surface discontinuities or orientation constraints. Grimson's algorithm was also plagued by an extremely slow convergence rate.

The primary contribution of Terzopoulos [1984] was improved computational efficiency of surface reconstruction. He used multigrid methods [Brandt 1977] to speed up his algorithm by orders of magnitude over the single-grid approach. Multigrid speedup ideas will be discussed in detail in Chapter 4. Terzopoulos addressed the full surface reconstruction problem since he integrated both depth and orientation constraints provided by various visual modalities. In addition, he studied techniques for the detection and the explicit representation of surface discontinuities. As mentioned in the beginning of this chapter, the

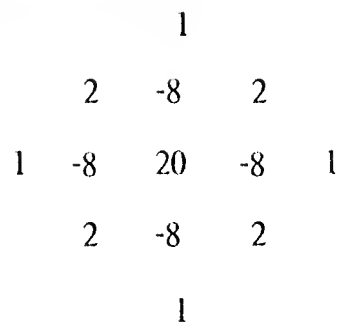


Figure 2-5. 2D Biharmonic operator.

location of discontinuities is assumed to be input to the surface reconstruction module. Some thoughts about how we might combine discontinuity detection with the coupled depth/slope network are discussed in Chapter 6.

Both Grimson and Terzopoulos used the standard discrete biharmonic operator for their surface reconstruction algorithms. Grimson used finite difference methods [1981, p. 180] to arrive at the biharmonic mask. Terzopoulos used more general finite element techniques but chose simple uniform squares to produce the same biharmonic mask. The biharmonic mask, pictured in Figure 2-5, can be viewed as a constraint equation relating neighboring depth points. The mask can equivalently be expressed as

$$\begin{aligned} & 20u_{i,j} \\ & - 8(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}) \\ & + 2(u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i+1,j+1}) \\ & + (u_{i-2,j} + u_{i+2,j} + u_{i,j-2} + u_{i,j+2}) = 0 \end{aligned} \tag{2.5}$$

The large number of relations of this type form an enormous system of linear equations to be solved. Many solution techniques are available for solving linear systems and they fall logically into two main categories: direct and iterative methods. Direct methods take a finite number of steps to come up with an answer. Iterative methods start with some initial guess of the solution and

gradually converge. Iterative methods are preferred for large problems such as this because they are faster, easier to program and take less storage than the corresponding direct methods. Grimson used a gradient projection iterative algorithm while Terzopoulos chose the Gauss-Seidel iterative method on each grid in a hierarchy of grids to speed convergence.

The Gauss-Seidel method is the sequential replacement of each point in the mesh by some function of other points in the mesh. One Gauss-Seidel realization of (2.5) is:

$$\begin{aligned} u_{i,j}^{k+1} = \frac{1}{20}\big[8(&u_{i-1,j}^{k+1} + u_{i+1,j}^{k} + u_{i,j-1}^{k+1} + u_{i,j+1}^{k}) \\ &-2(u_{i-1,j-1}^{k+1} + u_{i-1,j+1}^{k} + u_{i+1,j-1}^{k+1} + u_{i+1,j+1}^{k}) \\ &-(u_{i-2,j}^{k+1} + u_{i+2,j}^{k} + u_{i,j-2}^{k+1} + u_{i,j+2}^{k})\big] \end{aligned}$$

Notice that each node replaces itself by a function of its neighbors some of them old and some of them new. For this reason, the Gauss-Seidel scheme is inherently sequential. A Jacobi iterative scheme calls for the replacement of all node values simultaneously, and so can be parallel. The simple extension of the Gauss-Seidel relaxation to the fully parallel Jacobi scheme results in:

$$\begin{aligned} u_{i,j}^{k+1} = \frac{1}{20}\big[8(&u_{i-1,j}^{k} + u_{i+1,j}^{k} + u_{i,j-1}^{k} + u_{i,j+1}^{k}) \\ &-2(u_{i-1,j-1}^{k} + u_{i-1,j+1}^{k} + u_{i+1,j-1}^{k} + u_{i+1,j+1}^{k}) \\ &-(u_{i-2,j}^{k} + u_{i+2,j}^{k} + u_{i,j-2}^{k} + u_{i,j+2}^{k})\big] \end{aligned} \qquad (2.6)$$

Unfortunately, the standard biharmonic mask is unstable when used in a parallel Jacobi iteration. See Figure 2-6 for an example of this instability. The mathematical reasons for the instability of the parallel biharmonic mask are well known. See [Varga 1962; Young 1971] for more detail.

Brandt [1977] describes a technique called the weighting of corrections to obtain convergence for the parallel biharmonic operator. A single iteration consists of two steps: a biharmonic application and a local weighting. In the first

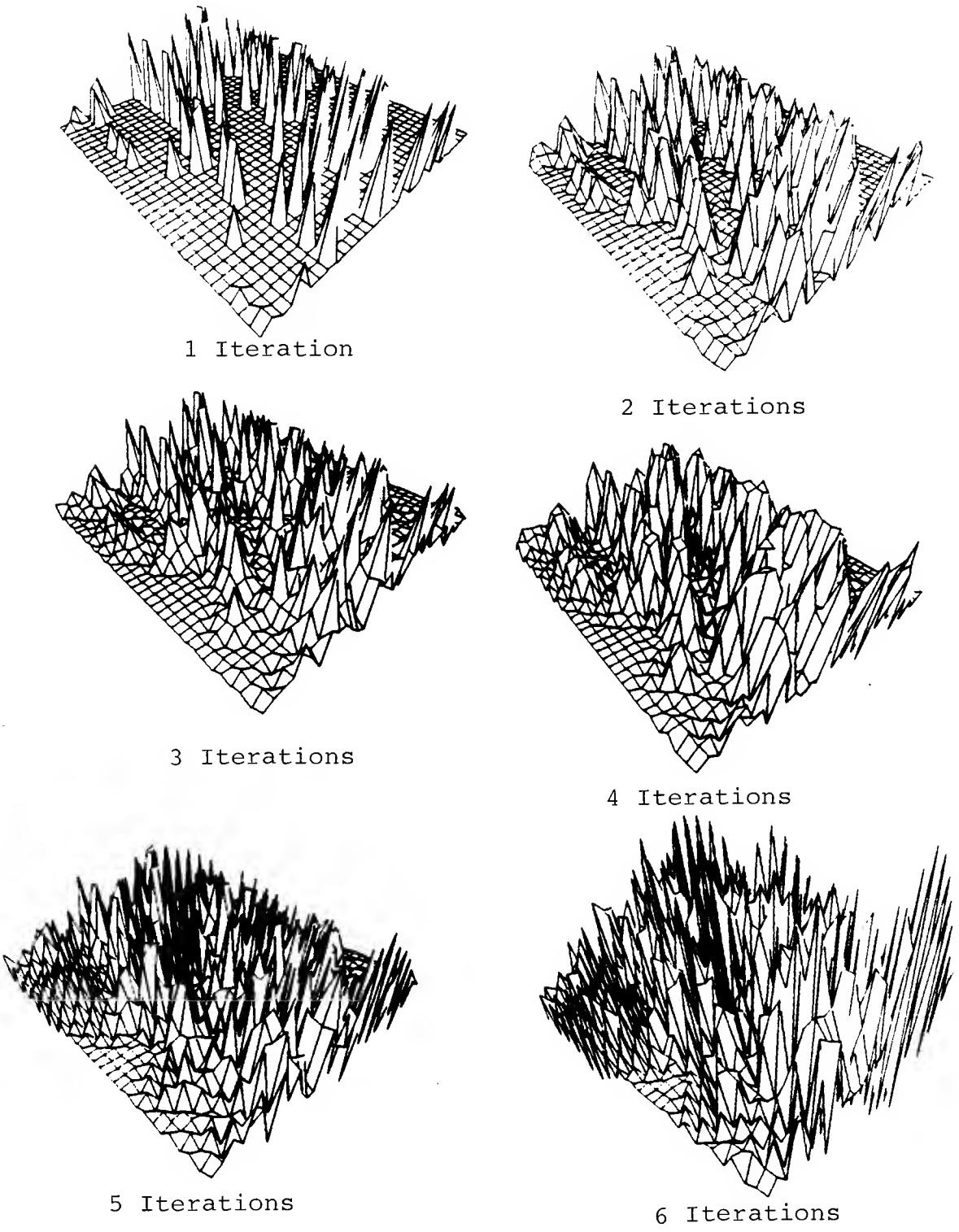


Figure 2-6. Jacobi algorithm diverging for a 2D problem.

step we calculate the required correction, $\delta u^k_{i,j}$, given by the Jacobi biharmonic replacement (2.6), where $u^k + \delta^k = u^{k+1}$. That is,

$$\delta u^k_{i,j} = -u^k_{i,j} + \frac{1}{20}\big[8(u^k_{i-1,j} + u^k_{i+1,j} + u^k_{i,j-1} + u^k_{i,j+1}) \\ -2(u^k_{i-1,j-1} + u^k_{i-1,j+1} + u^k_{i+1,j-1} + u^k_{i+1,j+1}) \\ -(u^k_{i-2,j} + u^k_{i+2,j} + u^k_{i,j-2} + u^k_{i,j+2})\big]$$

We do not use $\delta u_{i,j}$ as the correction term since the method does not converge. Instead we choose some linear combination of the local $\delta u_{i,j}$ as the correction term. In particular, Brandt suggests:

$$u^{k+1}_{i,j} = u^k_{i,j} + \omega_\alpha(\delta u^k_{i,j}) + \omega_\beta(\delta u^k_{i-1,j} + \delta u^k_{i+1,j} + \delta u^k_{i,j-1} + \delta u^k_{i,j+1})$$

For the optimal smoothing rate Brandt derived:

$$\omega_\alpha = 1.552$$

$$\omega_\beta = 0.353$$

Brandt points out that this technique is very sensitive to changes. For example, if ω_α were changed to 1.4 the method would not converge. In one sense, this weighting scheme is equivalent to the application of a 25-point operator. That is, a single iteration step at each point in the mesh requires the values of 24 neighboring points.

There is no natural way to integrate orientation constraints short of going back to the original energy functional and adding some kind of penalty term. In fact this is the method used by Terzopoulos [1984]. At points where the orientation is known, he creates a new energy term which is the square of the difference between the orientation measurement and the explicitly calculated slope. Terzopoulos adds this term into his energy functional with some appropriately chosen weighting factor.

Chapter 3 develops the coupled depth/slope model. This new approach to surface reconstruction leads to a digital algorithm which has a smaller mask size

than the weighted technique discussed above. Also, orientation and discontinuity information are straightforwardly integrated into the surface description.

CHAPTER 3
THE COUPLED DEPTH/SLOPE MODEL

The coupled depth/slope model is a network of ideal subtractor elements connected by two planes of resistor meshes. This novel model, based upon analog network considerations, has significant advantages over previous models of surface reconstruction.

3.1. 1D Form of the Coupled Depth/Slope Model

For simplicity, we first discuss the model in one dimension. The proposed network consists of depth nodes connected to slope nodes by ideal subtract constraint boxes. The continuous 1D network is shown in Figure 3-1. A resistor mesh is used to smooth the slope points. The voltage at the u nodes represent actual depths and the the values of p correspond to slope. Note that u and p are continuous functions of x. This model is called the coupled depth/slope model because of the coupling between the depth and slope representations provided by the ideal subtractor elements. The subtractors explicitly calculate a slope representation of the surface. We smooth out any sudden changes in the slope with a resistor mesh and pass the information back up to the depth plane. Any depth or slope node can be made into a constraint by fixing an ideal voltage source to the proper place in the network. For now, assume that the data are exact, forcing the reconstructed depth and slope values to pass through the given constraint values. Noisy and conflicting data are discussed in section 3.3.

The subtractor device is shown in Figure 3-2. If nodes A and B are set with ideal voltage sources then node C will be set to $A - B$ by the device. The device

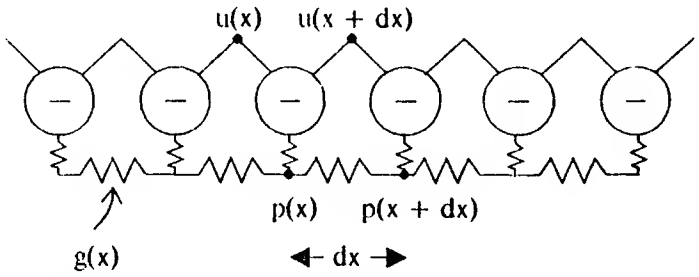


Figure 3-1. 1D network for coupled depth/slope surface reconstruction

is unusual in that all of its terminals can act as inputs or outputs. If nodes B and C are held constant with voltage sources, then the A terminal is fixed to $B + C$. If A and C are input, then B becomes $A - C$.

Now we show that this coupled model has the right properties for surface reconstruction as defined in Chapter 2. The model must find a smooth surface which fits the given depth constraints. In addition, orientation constraints and surface discontinuities must be incorporated into the final surface description.

First, we show that the network provides for biharmonic smoothness, used by Grimson and Terzopoulos, when depth constraints only are provided. This is shown by first writing the power, E, dissipated in the network. The minimization of this power yields the biharmonic equation. The power dissipated in a single resistor equals the square of the voltage drop through the resistor times the conductance of the resistor ($E = gV^2$). We assume that the vertical resistors have unity conductance and the ideal subtractor elements consume negligible power. To find the total power dissipated we integrate over all x values.

$$E = \int (\frac{du}{dx} - p)^2 + g(x)(\frac{dp}{dx})^2 \, dx \qquad (3.1)$$

The first and second terms in the integral represent the power dissipated in the vertical and horizontal resistors of the network, respectively.

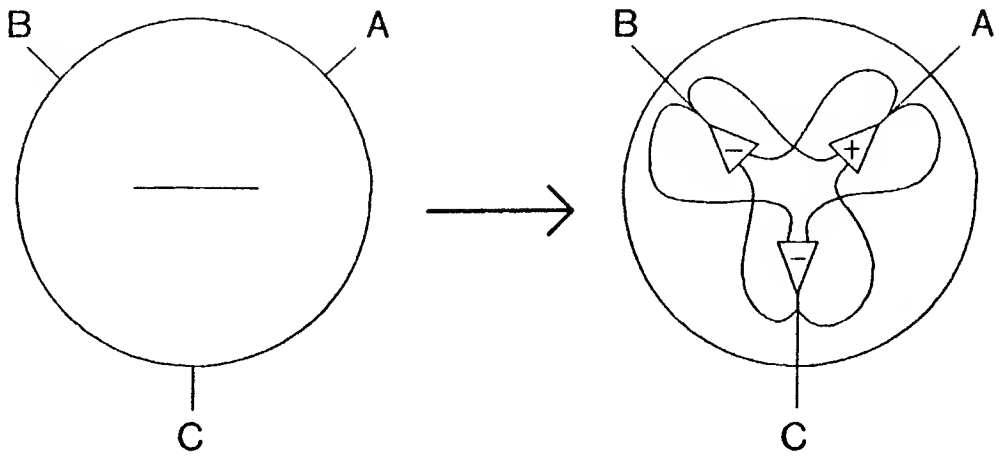


Figure 3-2. Tri-directional subtraction device.

Maxwell's minimum heat theorem states that the distribution of currents and voltages in a circuit is such that the total power dissipated as heat is minimized. The Euler-Lagrange equation are used to find the minimum energy state reached by the network. The appropriate 1D Euler-Lagrange equations are:

$$\frac{d}{dx}E_{u_x} - E_u = 0 \tag{3.2}$$

$$\frac{d}{dx}E_{p_x} - E_p = 0 \tag{3.3}$$

Assume for now that $g(x) = 1$ for all x. Applying these Euler-Lagrange equations to find the stationary points of the energy functional, E, results in:

$$\frac{d^2u}{dx^2} = \frac{dp}{dx} \tag{3.4}$$

$$\frac{d^2p}{dx^2} = p - \frac{du}{dx} \tag{3.5}$$

These are a pair of coupled second order equations relating u and p. These equations are shown to reduce to the biharmonic equation by first taking $\frac{d^2}{dx^2}$ of (3.4) and $\frac{d}{dx}$ of (3.5) giving:

$$\frac{d^4u}{dx^4} = \frac{d^3p}{dx^3} \tag{3.6}$$

$$\frac{d^3p}{dx^3} = \frac{dp}{dx} - \frac{d^2u}{dx^2} \tag{3.7}$$

or

$$\frac{d^4u}{dx^4} = \frac{dp}{dx} - \frac{d^2u}{dx^2} \tag{3.8}$$

But we know that $\frac{dp}{dx} - \frac{d^2u}{dx^2} = 0$ from (3.4), so

$$\frac{d^4u}{dx^4} = 0 \tag{3.9}$$

This is the one-dimensional form of the biharmonic equation.

3.2. Similarities to the Human Visual System

There is a lot of redundancy in the $2\frac{1}{2}$-D sketch described here because both depth and slopes are explicitly represented. Only one of these measures need by stored, and the other can be recovered with a simple derivative or integral. Marr [1982, p. 279] draws on psychophysical evidence to argue that the human biological $2\frac{1}{2}$-D sketch makes explicit representations of orientation as well as depth.

3.3. Extending the Model to Handle Noisy Constraints

Thus far we have assumed that the given depth and orientation constraints are exact. We force the depth and slope planes to pass exactly through the constraint points. In reality, there are additive noise components for each constraint measurement. It would be unwise to force the reconstructed surface to pass through noisy, inaccurate points. This problem is solved by adding another term to the energy functional. We rewrite the energy in (3.1) as:

$$E = \int (u_x - p)^2 + (p_x)^2 + \alpha(x)(u - \hat{u})^2 + \beta(x)(p - \hat{p})^2 \, dx \tag{3.10}$$

The $\hat{u}$ and $\hat{p}$ are the values of the depth and slope measurements respectively. The variables $\alpha(x)$ and $\beta(x)$ represent the confidence of the measured values in depth and slope. For example, if we knew that the measurement $\hat{u}(x_0)$ was inaccurate we could set the corresponding $\alpha(x_0)$ to zero.

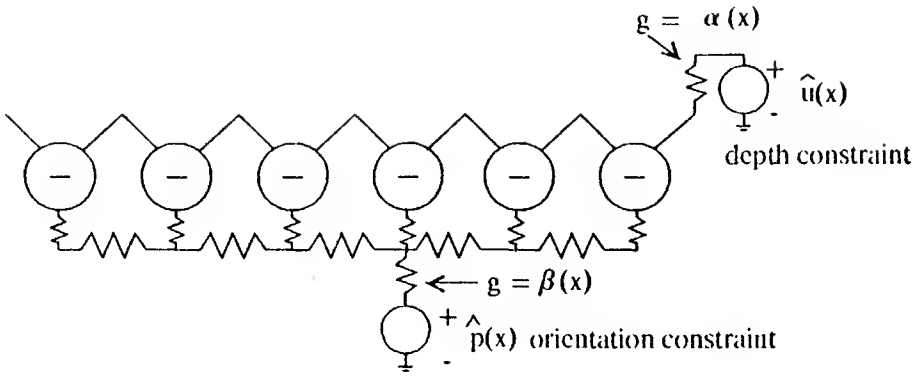


Figure 3-3. Handling noisy constraints.

Similar energy terms were used by Terzopoulos, Grimson and Marroquin. According to Marroquin [1985],

$$\alpha(x) = \frac{1}{2\sigma^2(\hat{u})} \tag{3.11}$$

$$\beta(x) = \frac{1}{2\sigma^2(\hat{p})} \tag{3.12}$$

where $\sigma(\hat{u})$ represents the standard deviation of zero mean, white, additive Gaussian noise of the measurement $\hat{u}$. Therefore $\alpha(x)$ and $\beta(x)$ represent a tradeoff between surface smoothness and how close the surface fits to the depth and orientation constraints.

The added noise term can be modeled as a conductance times the square of a voltage and so is easily added to our electrical model. We now set constraints with a voltage source through a series resistor to the constraint point in the network. These connections are shown in Figure 3-3. The voltage source has a value of the depth measurement $\hat{u}$. This voltage is applied through a resistor with conductance $\alpha(x)$, where $\alpha(x)$ is related to the amount of noise in $\hat{u}$ by equation (3.11). If there is no depth constraint at a particular location, say x_0, we choose an arbitrary constraint value $\hat{u}(x_0)$ for the voltage source with an

open circuit, $\alpha(x_0) = 0$. This is equivalent to saying that there is infinite noise in $\hat{u}(x_0)$ and that we have no confidence in that surface measurement.

Minimizing (3.10) gives:

$$\frac{d^2u}{dx^2} = \frac{dp}{dx} + \alpha(u - \hat{u}) \tag{3.13}$$

$$\frac{d^2p}{dx^2} = p - \frac{du}{dx} + \beta(p - \hat{p}) \tag{3.14}$$

The solution of these equations gives a smooth surface which performs a least squares averaging reduction of noise. A very different method for handling noise is proposed in Chapter 4. In a digital algorithm, we may want to assume noiseless measurements and solve the problem exactly. Then we can perform some amount of smoothing dependent on how much noise is present. This post smoothing step is very fast since we are dealing with dense surface data.

3.4. Generalized Smoothness

How much smoothness should we require of a surface? Some surfaces are very jagged and the biharmonic reconstructs a surface that is much too smooth. On the other hand, some surfaces are even more smooth than the biharmonic; they may have smooth second or third derivatives. The biharmonic is typically used for surface interpolation because it seems to do reasonably well over all possible surfaces.

Terzopoulos [1986] proposes Tikhonov stabilizers [Tikhonov 1977] to generalize smoothness to any order of derivative. For example, the 1D Tikhonov stabilizer appropriate for our problem is:

$$E = \sum_{m=0}^{p} \int g_m(x) \left(\frac{d^m u(x)}{dx^m}\right)^2 dx \tag{3.15}$$

where the $g_m(x)$ are continuous weighting functions and $\frac{d^m u(x)}{dx^m}$ is the m^{th} derivative of u. The generalized coupled depth/slope network directly implements

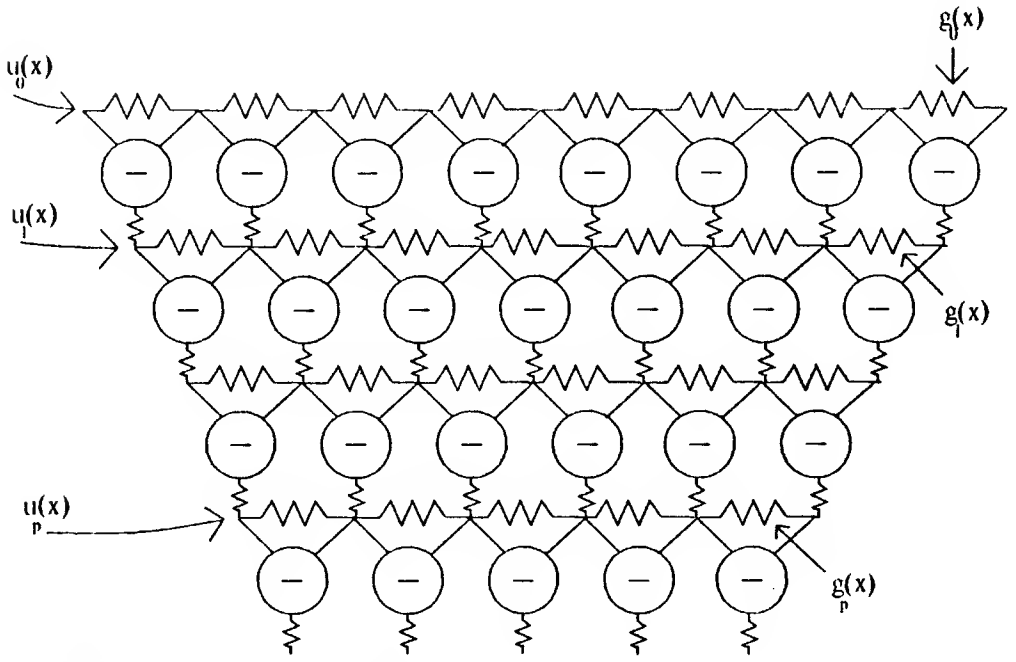


Figure 3-4. 1D coupled depth/slope model for generalized smoothness

the controlled-continuity stabilizers of Terzopoulos. The generalized network is shown in Figure 3-4. Here $u_0(x)$ represents the depth values, while $u_1(x)$ represents the first derivative of $u_0(x)$. Naturally, $u_m(x)$ represents the m^{th} derivative function of $u_0(x)$, and $g_m(x)$ denotes the conductance function of level m of the network.

As before, we can write the energy dissipated in all the resistors in the network as:

$$E = \sum_{m=o}^{p} \int \left[g_m(x) \left(\frac{du_m(x)}{dx} \right)^2 + \left(\frac{d}{dx} u_m(x) - u_{m+1}(x) \right)^2 \right] dx \qquad (3.16)$$

The first term in the integral represents the power dissipated in the horizontal resistors on the m^{th} level, and the second term is the power lost in the vertical resistors.

This very general network can incorporate constraints of any order of derivative into the surface solution. This is done simply by setting the proper mesh points with a voltage source. Also, discontinuities in the m^{th} order derivative at

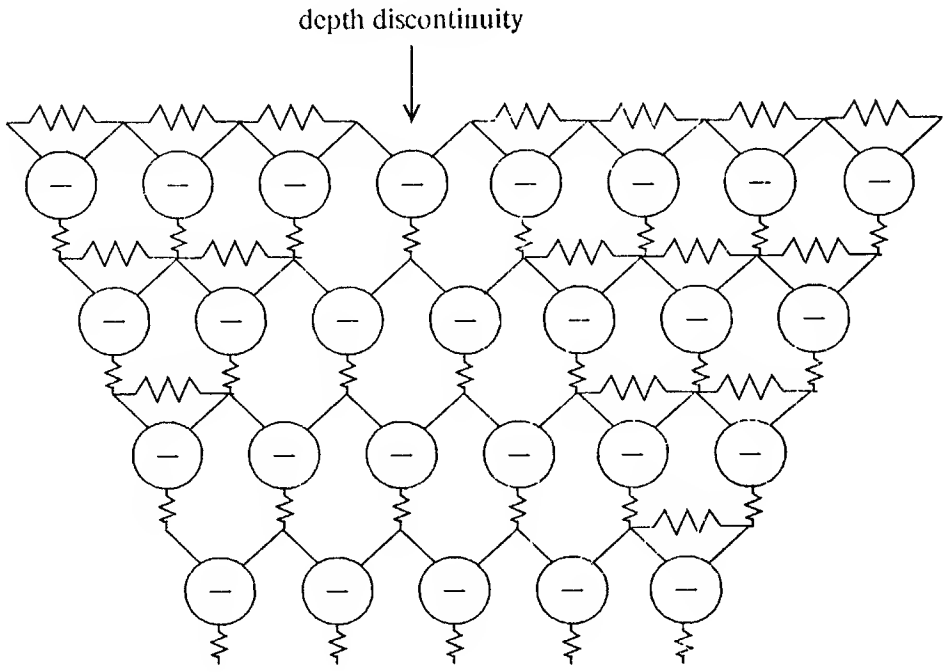


Figure 3-5. A depth discontinuity.

$x = x_0$ can be incorporated by setting $g_m(x_0) = 0$. For example, a depth discontinuity can be implemented by such a circuit as shown in Figure 3-5. Here, we have broken the resistors that smooth the first (and higher) derivatives. Depth discontinuities occur in between points in the u mesh.

Terzopoulos handles orientation discontinuities by switching to a membrane model instead of the thin plate at the location of the discontinuity. This allows the reconstructed surface to crease sharply but still remain continuous. In the coupled depth/slope model, membrane smoothness is provided by the top resistor mesh, which smoothes the depth values. The resistors on all other levels are broken. Orientation discontinuities occur exactly on u mesh locations. An example of an orientation discontinuity is shown in Figure 3-6. The thin plate model is not rigid enough to allow other even higher order discontinuities, but the coupled depth/slope network can handle discontinuities of any order.

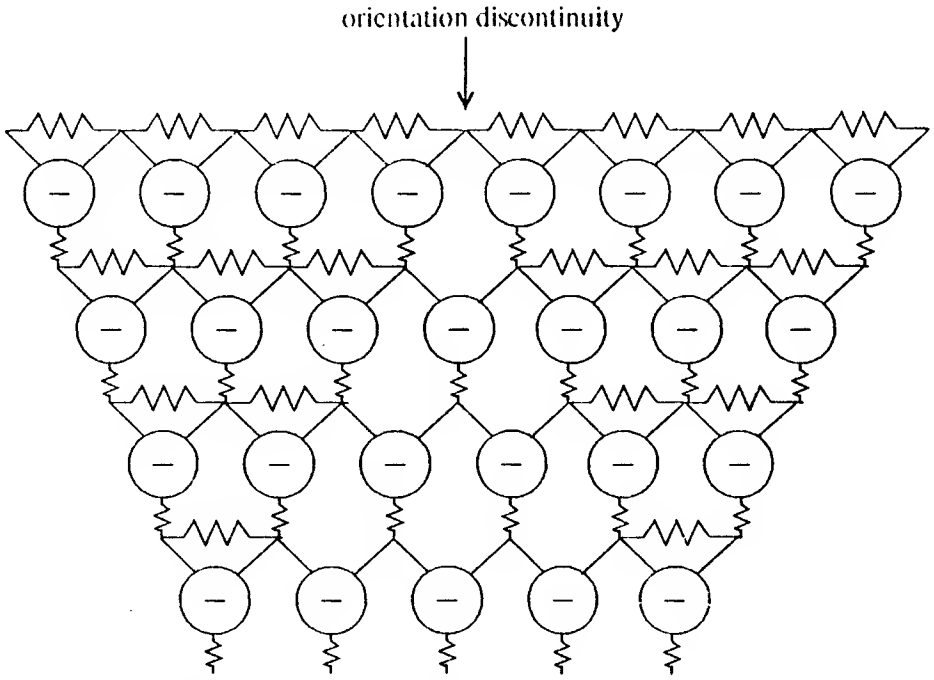


Figure 3-6. An orientation discontinuity.

For some surfaces, the thin plate solution is too smooth. However, we cannot use the membrane solution because of its tendency to crease and its poor extrapolation properties. Splines under tension were invented to combat a similar problem in graphics [Pilcher 1974]. Splines under tension provide a mechanism for combining varying amounts of smoothness from the thin plate and membrane smoothness. They are also proposed as a surface interpolation technique to reconstruct surfaces with intermediate smoothness between thin plates and membranes [Terzopoulos 1984]. The coupled depth/slope network generalizes splines under tension since the network can integrate arbitrary combinations of membrane, thin plate and any higher order smoothness.

The general coupled depth/slope network generalizes the surface reconstruction algorithm of Terzopoulos [1984]. Information at arbitrary orders of smoothness can be usefully combined into the surface description.

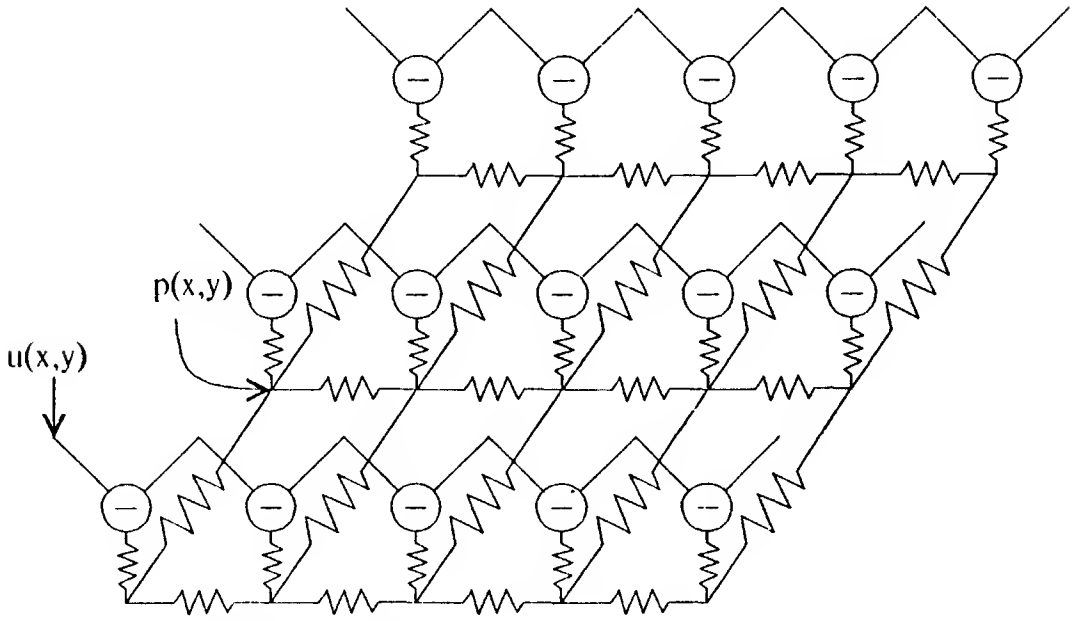


Figure 3-7. 2D extension – the p plane.

3.5. 2D Extension of the Coupled Depth/Slope Method

The 2D network for the coupled depth/slope method is shown in Figures 3-7 and 3-8. For simplicity, we revert back to biharmonic smoothness only, but the generalized smoothness ideas developed in the last section apply equally well in two dimensions. The 2D network is more complicated because the slopes in both the x and the y directions must be included. Therefore we use two resistor meshes to smooth the p and q slope planes, where p is the slope of u in the x direction and q is the slope in the y direction.

Again, the power dissipated in the total network is found to be:

$$E = \int\int \left[(u_x - p)^2 + (u_y - q)^2 + p_x^2 + p_y^2 + q_x^2 + q_y^2 \right] dx\, dy \qquad (3.17)$$

The extension for noisy constraints described in section 3.3 for the 1D case is also applicable here. The appropriate 2D Euler-Lagrange equations for (3.17)

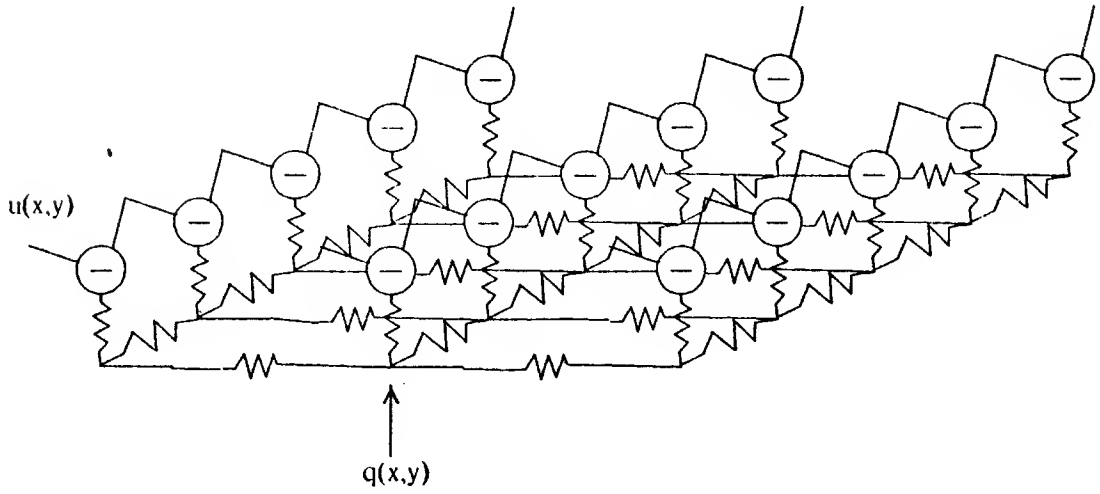


Figure 3-8. The q plane.

are:

$$\frac{\partial}{\partial x} E_{u_x} + \frac{\partial}{\partial y} E_{u_y} - E_u = 0$$
$$\frac{\partial}{\partial x} E_{p_x} + \frac{\partial}{\partial y} E_{p_y} - E_p = 0$$
$$\frac{\partial}{\partial x} E_{q_x} + \frac{\partial}{\partial y} E_{q_y} - E_q = 0$$

Applying these Euler-Lagrange equations to the energy function results in the following system of coupled differential equations:

$$\nabla^2 u = p_x + q_y \tag{3.18}$$

$$\nabla^2 p = p - u_x \tag{3.19}$$

$$\nabla^2 q = q - u_y \tag{3.20}$$

where $\nabla^2 = \left(\frac{\partial}{\partial x}\right)^2 + \left(\frac{\partial}{\partial y}\right)^2$.

Taking ∇^2 of (3.18) and $\frac{\partial}{\partial x}$ of (3.19) and $\frac{\partial}{\partial y}$ of (3.20) gives:

$$\nabla^2\nabla^2 u = \nabla^2 p_x + \nabla^2 q_y \tag{3.21}$$

$$\nabla^2 p_x = p_x - u_{xx} \tag{3.22}$$

$$\nabla^2 q_y = q_y - u_{yy} \tag{3.23}$$

Substituting (3.22) and (3.23) into (3.21) yields:

$$\nabla^2\nabla^2 u = p_x + q_y - u_{xx} - u_{yy} = p_x + q_y - \nabla^2 u$$

We know from equation (3.18) that $p_x + q_y - \nabla^2 u = 0$, so

$$\nabla^2\nabla^2 u = 0 \tag{3.24}$$

which is the two-dimensional biharmonic equation.

From the coupled differential equations, we cannot show that p and q must converge to u_x and u_y. A simple combination of the original three coupled equations gives:

$$\nabla^2\nabla^2 u = \frac{\partial}{\partial x}(p - u_x) + \frac{\partial}{\partial y}(q - u_y) = 0$$

We can only show that $u_x = p$ and $u_y = q$ if some boundary conditions are assumed or else we define $p_y = q_x$.

3.6. Other Computational Models

Computational models which solve the biharmonic equation must necessarily contain some active devices. It can be shown that the biharmonic mask

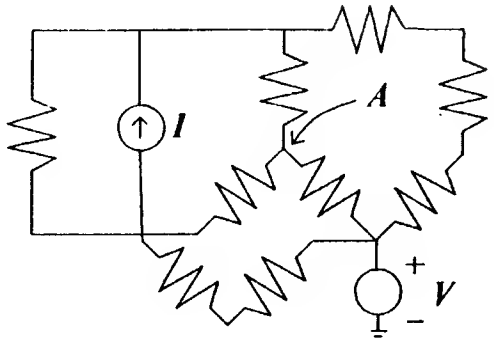


Figure 3-9. Arbitrary configuration of resistors and sources.

cannot be synthesized out of purely positive valued resistive components. The extrapolation mechanism required for biharmonic smoothness can boost the voltage on the exteriors of the array to higher and lower voltages than that which appear interior to the array. The resistor network alone cannot boost a voltage at a node higher than the voltage of any of the sources in the system. This is seen in Figure 3-9. The voltage at node A, for example, cannot be higher than the voltage at its neighboring nodes. If this were the case, than all currents will flow out of node A, violating Kirchoff's Current Law (KCL).

Nonetheless, there are applications for the simple membrane solution given by a mesh of equal valued resistors. For example, controlled industrial environments consisting of flat, zero-slope planes, problems with undetectable discontinuities, and scenes with no discontinuities are all good applications for membrane smoothness. However, we should not attempt to tackle the full vision problem without a more rigid smoothing technique. In general, the membrane solution gives a fast, first approximation to the surface.

Another way to solve the biharmonic equation is to split it into two Poisson equations and solve them sequentially. It will be shown that this procedure only results in an approximate solution to our surface interpolation problem.

For example, the biharmonic equation is given by:

$$\nabla^4 u = f \mapsto \nabla^2 \nabla^2 u = f \tag{3.25}$$

The two poisson equations are as follows:

$$\nabla^2 v = f \tag{3.26}$$

$$\nabla^2 u = v \tag{3.27}$$

We solve for v in (3.26) and then use the result to solve for u in (3.27).

An actual implementation of this method shows that the solution of the first equation is slow compared to the solution of the second. This is because we are solving a very sparse system in the first equation. The solution of the second equation merely results in local perturbations of the dense values of v. Since the values of v are equally spaced, we could solve the second equation with a simple gaussian convolution step. The convolution method has been implemented since it has an additional advantage: any amount of smoothness can be integrated into the surface by varying the size of the Gaussian operator. A very narrow Gaussian will have little effect on the surface and so return membrane smoothness. A wider Gaussian will return thin plate and smoother surfaces. Using either method, the time it takes to solve the first equation is still much greater than the time to solve the second.

Thus, the dual Poisson method is another way to generate a quick, crude solution to the surface interpolation problem. This method improves upon the membrane solution by forcing the surface to be smooth without too much extra computation. The main difficulty with the dual-Poisson approach is the boundary conditions. Either the slope or the depth values must be specified around the boundary.

3.7. Summary

In summary, we have developed the coupled depth/slope model for surface

reconstruction. This is a computational model which has a number of features:

1. If depths only are given, the model reconstructs a surface with the same biharmonic smoothness used by Terzopoulos and Grimson. In addition, in the process of computing the depths, a dense slope representation is also calculated. For this reason, orientation constraints can be incorporated into the surface description with no extra work.

2. Like Terzopoulos' model, discontinuities in depth and orientation can be used to compute the reconstructed surface. However, the coupled depth/slope model generalizes to handle constraints of any order derivative. In addition, the model can handle discontinuities at any order of derivative of the surface.

3. The model seems to be biologically feasible. It fits the four criteria given in Chapter 1: parallelism, uniformity, locality and fault tolerance. The model's biharmonic smoothness properties are observed in psychophysical tests of humans. Also, Marr argues that the human $2\frac{1}{2}$-D sketch redundantly represents both slopes and depths. This compares favorably with the dense depth and slope representations generated by the coupled depth/slope network.

4. Finally, both digital and analog implementations of the coupled depth/slope method are possible.

CHAPTER 4

DIGITAL IMPLEMENTATION

This chapter develops a digital algorithm based upon the coupled depth/slope model. Again, we start with the 1D case before we progress to the 2D version. We apply standard finite difference methods to the continuous differential equations derived in the last chapter to get a large system of linear equations, which we solve with a parallel iterative algorithm. It is shown that this iterative algorithm is very well suited to a mesh-of-processors computer such as the Connection Machine.

4.1. The Discrete 1D Case

The coupled differential equations for u and p derived in Chapter 3 were:

$$\frac{d^2u}{dx^2} = \frac{dp}{dx} \tag{3.4}$$

$$\frac{d^2p}{dx^2} = p - \frac{du}{dx} \tag{3.5}$$

Applying simple finite difference approximations for first and second derivatives [Smith 1978, p. 6], as illustrated in Figure 4-1 yields:

$$u_{i-1} - 2u_i + u_{i+1} = p_i - p_{i-1}$$

$$p_{i-1} - 2p_i + p_{i+1} = p_i - (u_{i+1} - u_i)$$

Rearranging to solve for u_i and p_i gives:

$$u_i = \frac{1}{2}\left[(u_{i-1} + p_{i-1}) + (u_{i+1} - p_i)\right] \tag{4.1}$$

$$p_i = \frac{1}{3}\left[(u_{i+1} - u_i) + p_{i-1} + p_{i+1}\right] \tag{4.2}$$

A full Jacobi iteration method (simultaneous displacement) for these equations fails to converge to a solution. A proof of this divergence for the 2D case will

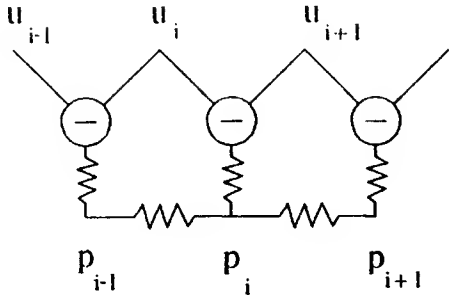


Figure 4-1. Discrete coupled depth/slope model.

be shown in section 4.2. However, a simple alternating process of first calculating the new depth values and then the new slope values guarantees convergence. The iteration method is therefore:

$$u_i^{k+1} = \frac{1}{2}\left[(u_{i-1}^k + p_{i-1}^k) + (u_{i+1}^k - p_i^k)\right] \tag{4.3}$$

$$p_i^{k+1} = \frac{1}{3}\left[(u_{i+1}^{k+1} - u_i^{k+1}) + p_{i-1}^k + p_{i+1}^k\right] \tag{4.4}$$

where the superscript k denote the iteration step. A proof of convergence for the 2D the alternating depth/slope algorithm is given in section 4.2.

4.2. The 2D Discrete Case

We would now like to extend our analysis to two-dimensions. We know from Chapter 3,

$$\nabla^2 u = p_x + q_y \tag{3.16}$$

$$\nabla^2 p = p - u_x \tag{3.17}$$

$$\nabla^2 q = q - u_y \tag{3.18}$$

Applying 2D finite difference approximations to the coupled differential equations and rearranging terms results in the following discrete equations:

$$u_{i,j} = \frac{1}{4}[(u_{i-1,j} + p_{i-1,j}) + (u_{i,j-1} + q_{i,j-1}) + (u_{i,j+1} - q_{i,j}) + (u_{i+1,j} - p_{i,j})]$$

$$p_{i,j} = \frac{1}{5}[(u_{i+1,j} - u_{i,j}) + p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1}]$$

$$q_{i,j} = \frac{1}{5}[(u_{i,j+1} - u_{i,j}) + q_{i+1,j} + q_{i-1,j} + q_{i,j+1} + q_{i,j-1}] \qquad (4.5)$$

These finite difference equations make up a large but sparse system of equations that must be solved. Because of the size of the system, iterative methods are generally used. Standard iterative techniques will be used, but other alternatives are discussed later in the chapter.

The discretized equations for u, p and q derived above map directly into a Gauss-Seidel scheme for a conventional sequential computer. This direction, though important, is not developed. We will concentrate on Jacobi iterative schemes which are suitable for fine grain, parallel computers. The architectures of primary concern are locally connected meshes of very simple processing elements such as the Connection Machine [Hillis 1985].

These architectures have a number of features in common. They all consist of a 2D rectangular mesh of tens of thousands of extremely simple processing elements. Inter-processor communication is performed through nearest neighbor connections only. Machines such as the Connection Machine have special hardware for arbitrary global communication but this communication is much slower than local communication and is not appropriate for any of the algorithms developed in this report. The processing elements typically contain only a one-bit ALU. This necessarily requires that all arithmetic be performed bit serially. Since many processors are packaged per chip, each processing element is allotted only a limited amount of on chip memory.

The algorithms developed in this chapter have been devised to match the architecture constraints of machines such as the Connection Machine, so that

surfaces can be calculated as fast as possible. Thus the algorithms exhibit local operations which can be performed in parallel over all points in the mesh, can be calculated easily with fixed point arithmetic, and require very little memory.

A first approach at developing a Jacobi iterative scheme can be constructed from the coupled difference equations (4.5):

$$u_{i,j}^{k+1} = \frac{1}{4}[(u_{i-1,j}^{k} + p_{i-1,j}^{k}) + (u_{i,j-1}^{k} + q_{i,j-1}^{k}) + (u_{i,j+1}^{k} - q_{i,j}^{k}) + (u_{i+1,j}^{k} - p_{i,j}^{k})]$$

$$p_{i,j}^{k+1} = \frac{1}{5}[(u_{i+1,j}^{k} - u_{i,j}^{k}) + p_{i+1,j}^{k} + p_{i-1,j}^{k} + p_{i,j+1}^{k} + p_{i,j-1}^{k}]$$

$$q_{i,j}^{k+1} = \frac{1}{5}[(u_{i,j+1}^{k} - u_{i,j}^{k}) + q_{i+1,j}^{k} + q_{i-1,j}^{k} + q_{i,j+1}^{k} + q_{i,j-1}^{k}] \qquad (4.6)$$

Like its 1D counterpart, this method does not converge to a solution. This should not be surprising since it has been shown that this method solves the biharmonic equation for depths and that the Jacobi biharmonic iteration does not converge. Local Fourier analysis is now be used to show rigorously that the above iteration method does not converge.

First, we define the error values Δu^k, Δp^k, and Δq^k as the differences between the values of u, p, and q at a given iteration step k, and the final values of the true solution.

$$\Delta u_{i,j}^{k} \equiv u_{i,j} - u_{i,j}^{k}$$

$$\Delta p_{i,j}^{k} \equiv p_{i,j} - p_{i,j}^{k}$$

$$\Delta q_{i,j}^{k} \equiv q_{i,j} - q_{i,j}^{k}$$

It is hoped that:

$$\lim_{k\to\infty} \Delta u_{i,j}^{k} = 0$$

$$\lim_{k\to\infty} \Delta p_{i,j}^{k} = 0$$

$$\lim_{k\to\infty} \Delta q_{i,j}^{k} = 0$$

Similarly we define the errors between the values at step $k+1$ And the final solution as:

$$\Delta u_{i,j}^{k+1} \equiv u_{i,j} - u_{i,j}^{k+1}$$
$$\Delta p_{i,j}^{k+1} \equiv p_{i,j} - p_{i,j}^{k+1}$$
$$\Delta q_{i,j}^{k+1} \equiv q_{i,j} - q_{i,j}^{k+1}$$

These equations can be expanded by taking the difference of equations (4.5) from (4.6):

$$\begin{aligned}
\Delta u_{i,j}^{k+1} &= \frac{1}{4}[(\Delta u_{i-1,j}^{k} + \Delta p_{i-1,j}^{k}) + (\Delta u_{i,j-1}^{k} + \Delta q_{i,j-1}^{k}) \\
&\quad + (\Delta u_{i,j+1}^{k} - \Delta q_{i,j}^{k}) + (\Delta u_{i+1,j}^{k} - \Delta p_{i,j}^{k})] \\
\Delta p_{i,j}^{k+1} &= \frac{1}{5}[(\Delta u_{i+1,j}^{k} - \Delta u_{i,j}^{k}) \\
&\quad + \Delta p_{i+1,j}^{k} + \Delta p_{i-1,j}^{k} + \Delta p_{i,j+1}^{k} + \Delta p_{i,j-1}^{k}] \\
\Delta q_{i,j}^{k+1} &= \frac{1}{5}[(\Delta u_{i,j+1}^{k} - \Delta u_{i,j}^{k}) \\
&\quad + \Delta q_{i+1,j}^{k} + \Delta q_{i-1,j}^{k} + \Delta q_{i,j+1}^{k} + \Delta q_{i,j-1}^{k}]
\end{aligned} \tag{4.7}$$

Now, let's assume that the error occurs with some Fourier spatial frequency (ω_1, ω_2) over the surface. We assume a different Fourier error for both the depth and slope values.

$$\Delta u_{i,j}^{k} = U^k e^{i(i\omega_1 + j\omega_2)}$$
$$\Delta p_{i,j}^{k} = P^k e^{i(i\omega_1 + j\omega_2)}$$
$$\Delta q_{i,j}^{k} = Q^k e^{i(i\omega_1 + j\omega_2)}$$

and

$$\Delta u_{i,j}^{k+1} = U^{k+1} e^{i(i\omega_1 + j\omega_2)}$$
$$\Delta p_{i,j}^{k+1} = P^{k+1} e^{i(i\omega_1 + j\omega_2)}$$
$$\Delta q_{i,j}^{k+1} = Q^{k+1} e^{i(i\omega_1 + j\omega_2)}$$

where U, P and Q are the magnitudes of the associated Fourier error. Substituting these expressions into the equation (4.7), dividing by $e^{i(i\omega_1 + j\omega_2)}$ and

rearranging terms gives:

$$\begin{pmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix} \begin{pmatrix} U \\ P \\ Q \end{pmatrix}^{k+1} = \begin{pmatrix} \frac{1}{2}f & \frac{1}{4}g & \frac{1}{4}h \\ \frac{1}{5}g & \frac{2}{5}f & 0 \\ \frac{1}{5}h & 0 & \frac{2}{5}f \end{pmatrix} \begin{pmatrix} U \\ P \\ Q \end{pmatrix}^{k}$$

where

$$f = cos(\omega_1) + cos(\omega_2)$$

$$g = e^{i\omega_1} - 1$$

$$h = e^{i\omega_2} - 1$$

The amplification matrix $A(\omega_1, \omega_2)$ shows how much the magnitude of the errors decreases for each iteration. $A(\omega_1, \omega_2)$ is defined as:

$$\begin{pmatrix} U \\ P \\ Q \end{pmatrix}^{k+1} = A(\omega_1, \omega_2) \begin{pmatrix} U \\ P \\ Q \end{pmatrix}^{k}$$

A simple calculation for $\omega_1 = \omega_2 = \pi$ yields eigenvalues greater than one in magnitude. This means that certain frequency components of the error are amplified during each iteration, preventing the calculation from converging.

However, we can add more stability to the iteration by requiring that we alternate between depth and slope values. That is, first we calculate all of the depths in parallel and then we calculate all of the slopes in parallel. This iteration scheme looks like the following:

$$u_{i,j}^{k+1} = \frac{1}{4}[(u_{i-1,j}^{k} + p_{i-1,j}^{k}) + (u_{i,j-1}^{k} + q_{i,j-1}^{k}) + (u_{i,j+1}^{k} - q_{i,j}^{k}) + (u_{i+1,j}^{k} - p_{i,j}^{k})]$$

$$p_{i,j}^{k+1} = \frac{1}{5}[(u_{i+1,j}^{k+1} - u_{i,j}^{k+1}) + p_{i+1,j}^{k} + p_{i-1,j}^{k} + p_{i,j+1}^{k} + p_{i,j-1}^{k}]$$

$$q_{i,j}^{k+1} = \frac{1}{5}[(u_{i,j+1}^{k+1} - u_{i,j}^{k+1}) + q_{i+1,j}^{k} + q_{i-1,j}^{k} + q_{i,j+1}^{k} + q_{i,j-1}^{k}]$$

Notice that the new u^{k+1} values are computed using old values of depths and slope (u^k, p^k, and q^k). However, the new p^{k+1} and q^{k+1} values need the newly computed values of the depth (u^{k+1}). There is an inherent sequential nature to the algorithm since the depths must be computed first and then the slopes.

Going through the same type of error analysis as above yields the Fourier error equation of:

$$\begin{pmatrix} 4 & 0 & 0 \\ -g & 5 & 0 \\ -h & 0 & 5 \end{pmatrix} \begin{pmatrix} U \\ P \\ Q \end{pmatrix}^{k+1} = \begin{pmatrix} \frac{1}{2}f & \frac{1}{4}g & \frac{1}{4}h \\ 0 & \frac{2}{5}f & 0 \\ 0 & 0 & \frac{2}{5}f \end{pmatrix} \begin{pmatrix} U \\ P \\ Q \end{pmatrix}^{k}$$

with f, g and h defined as above. This results in an amplification matrix $A(\omega_1, \omega_2)$ of

$$\begin{pmatrix} \frac{1}{2}f & \frac{1}{4}g & \frac{1}{4}h \\ \frac{1}{10}fg & \frac{2}{5}f + \frac{1}{20}g^2 & \frac{1}{20}gh \\ \frac{1}{10}fh & \frac{1}{20}gh & \frac{2}{5}f + \frac{1}{20}h^2 \end{pmatrix}$$

It has been shown numerically with the EISPACK routines [Goos and Hartmanis 1976] that the maximum eigenvalue of the amplification matrix for this case has a value of .953 at $\omega_1 = \omega_2 = 2.79$. This shows that the method converges but has a very high smoothing rate and so would not be appropriate for a standard multigrid implementation.

This algorithm fits nicely on a computer such as the Connection Machine. Assume for now, that there are enough processors to fit each $u_{i,j}$ value in each processor. The corresponding $p_{i,j}$ and $q_{i,j}$ values are also stored in the same processor that contains $u_{i,j}$. This mapping scheme is shown in Figure 4-3.

There are two key advantages of this scheme. First, we pay no penalty for sequentially computing all of the depths (in parallel of course) and then computing all of the slopes (again in parallel). We couldn't compute the depths and slopes simultaneously anyway since they are calculated using the same processors.

Secondly, the iterative scheme requires that processors communicate only with their nearest four neighbors. This is in direct contrast to a 13-point biharmonic mask implementation, where 12 neighboring values must be collected. To make the 13-point biharmonic operator converge, an extra local weighting

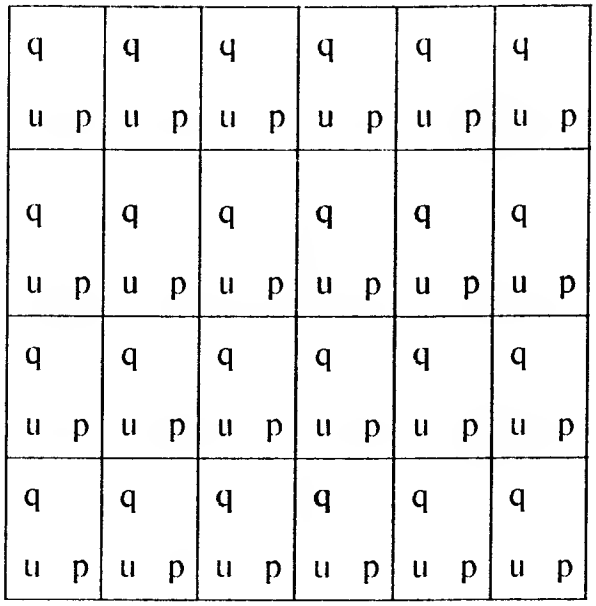


Figure 4-3. Mapping onto the Connection Machine

step must be added which turns each iteration into the equivalent of a 25-point operator application.

Another advantage of this scheme is that it requires no floating point operations. Floating point operations take thousands of cycles on processors with single bit ALUs typically found in these machines. These serial ALUs actually give these machines an advantage over conventional 16-bit or 32-bit computers. The inputs to the surface reconstruction module are fairly inaccurate and noisy. The most accuracy we should expect is about 8 bits. It is advantageous to only have to do 8-10 bit word arithmetic bit-serially. A conventional 32-bit computer gains nothing from this reduced word size.

4.3. Examples of Surface Reconstruction

The coupled depth/slope algorithm has been implemented and tested on

a number of different synthetic surfaces. Figure 4-4a shows some sparse data measurements that have been sampled from a synthetic surface. The surface reconstructed by the coupled depth/slope network is shown in Figure 4-4b. Figure 4-4c shows the result given the location of the circular discontinuity.

The next example, shown in 4-5 shows the reconstruction of a surface given only one u constraint and 100% of the p constraints. In effect, the coupled depth slope algorithm integrates the slope values and uses the single depth constraint to find the constant of integration.

Ikeuchi discusses a similar type problem in [Ikeuchi 1983]. Here, some shape from shading module has produced the p and q slopes everywhere in the image. Ikeuchi employs the following least squares energy functional shown to recover the surface:

$$E = \int\int (u_x - p)^2 + (u_y - q)^2 \, dx \, dy \tag{4.7}$$

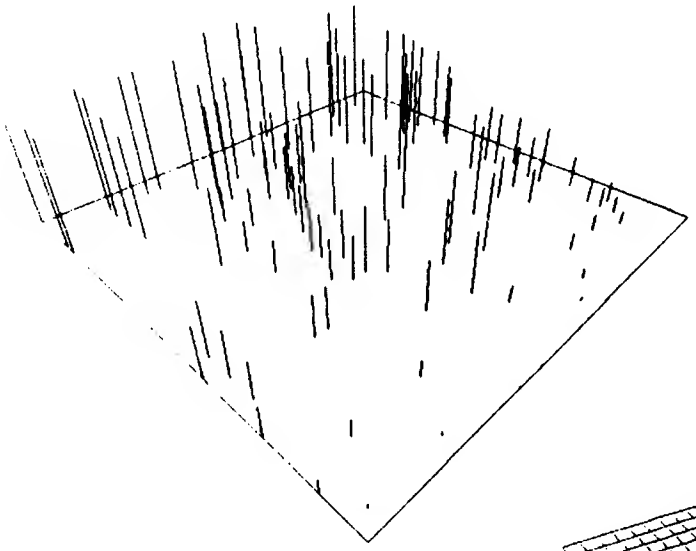
Minimizing this energy gives,

$$\nabla^2 u = p_x + q_y \tag{4.8}$$

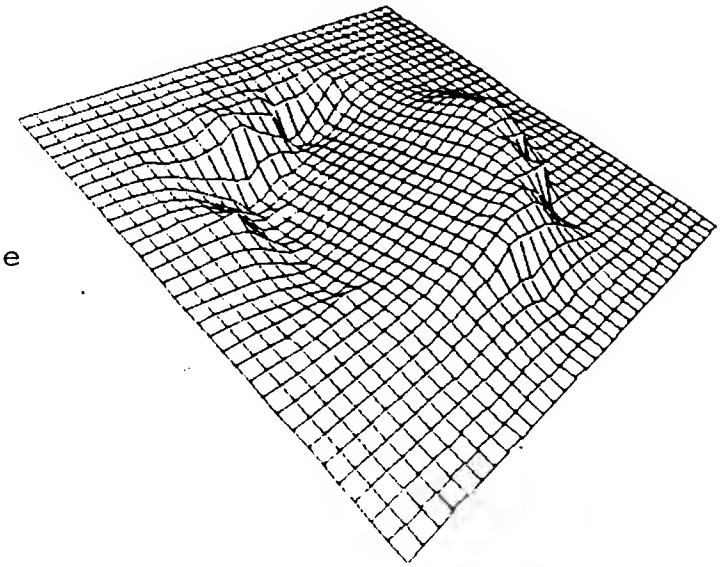
This leads to a simple iterative scheme to recover depth from the slope measurements. It should be noted that equation (4.8) derived by Ikeuchi is identical to the first of the three equations (3.18) describing the coupled depth/slope method. The coupled depth/slope method is therefore a generalization of the depth from shape algorithm developed by Ikeuchi.

How many constraints must be given in order for the coupled depth/slope problem to be well formulated? Terzopoulos [1984, p. 73] gives the minimal set of input constraints for the surface reconstruction problem to be well-posed as:

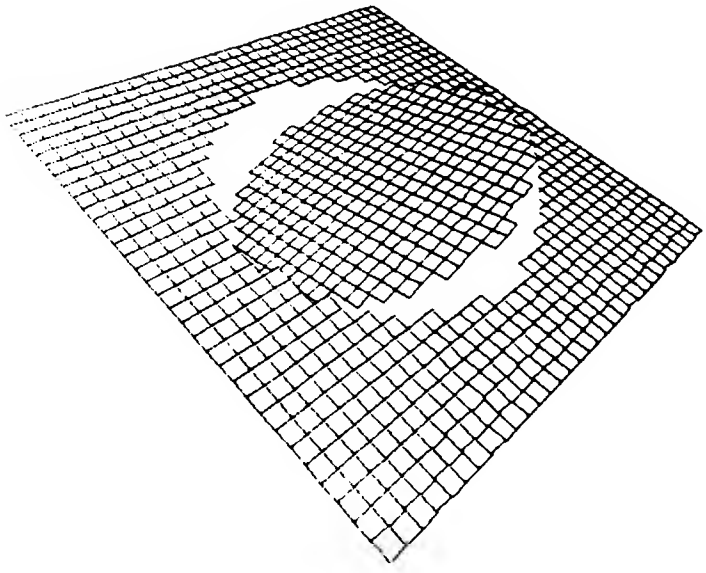
1. Three noncolinear depth constraints,
2. Two depth constraints as well as a single (nonaligned) p or q constraint,
3. A single depth constraint as well as a single p and a single q constraint.



a) 10% depth points



b) reconstructed surface



c) reconstructed surface with discontinuity

Figure 4-4. Example of surface reconstruction

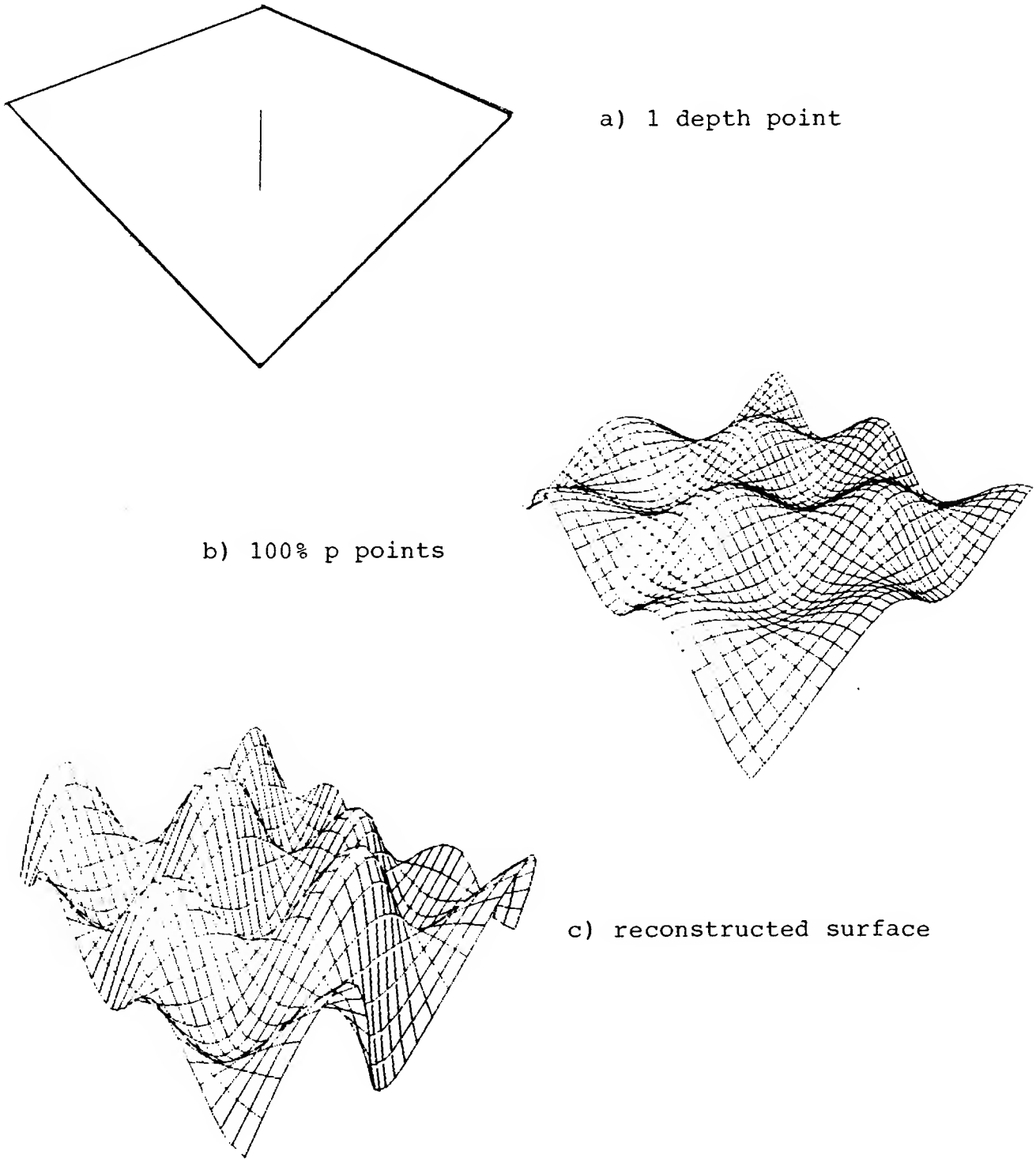


Figure 4-5. Reconstructing a surface given only p slopes

4.4. Extension for Handling Noisy Constraints

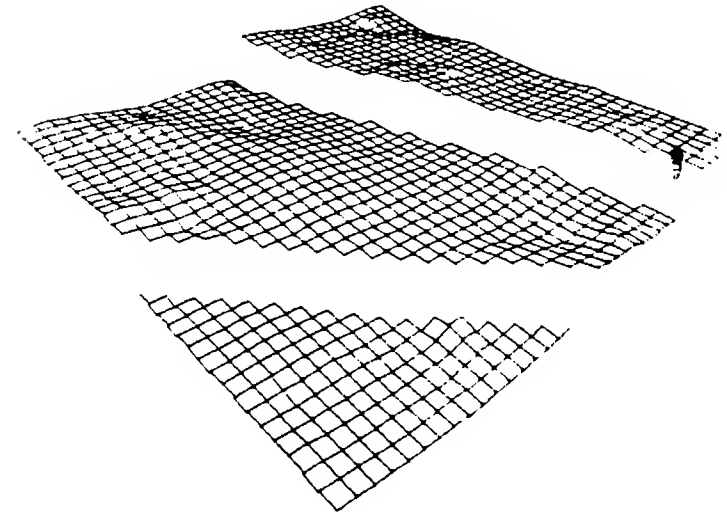
We now propose a different way for handling noise than that discussed in Chapter 3. In a digital algorithm on a Connection Machine, we assume noiseless measurements and solve the problem exactly. This keeps the critical loop which determines the speed of the algorithm as fast and simple as possible. After an exact solution is reached, we do some amount of smoothing dependent on how much noise is present in the constraint measurements. This post smoothing step is very fast since it deals with dense data.

An example of this post smoothing approach for handling noisy constraints is shown in Figure 4-6. Initially, the exact solution is very bumpy. We perform a number of smoothing steps proportional to how much noise we believe is in the data. These smoothing steps reduce the bumps in the surface. Different confidence values are handled by allowing the high noise points to change more than the low noise ones. This technique is similar but not equivalent to the least squares averaging approach used by Grimson and Terzopoulos.

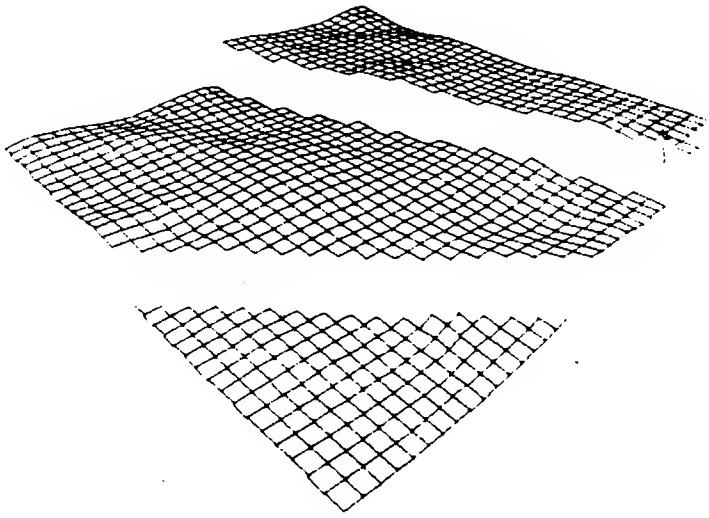
4.5. Fault Tolerance

Fault tolerance is a significant problem that has to be dealt with in massively parallel architectures. Suppose we fabricate a 32 x 32 array of processing elements on a wafer and that 10% of them are broken. Nevertheless, we would like the algorithm's performance to degrade gracefully as more processors become faulty. A diagnostic routine can be run in which processors check their neighbors to see if they are functioning properly. If a neighbor is faulty, a discontinuity is marked in that direction.

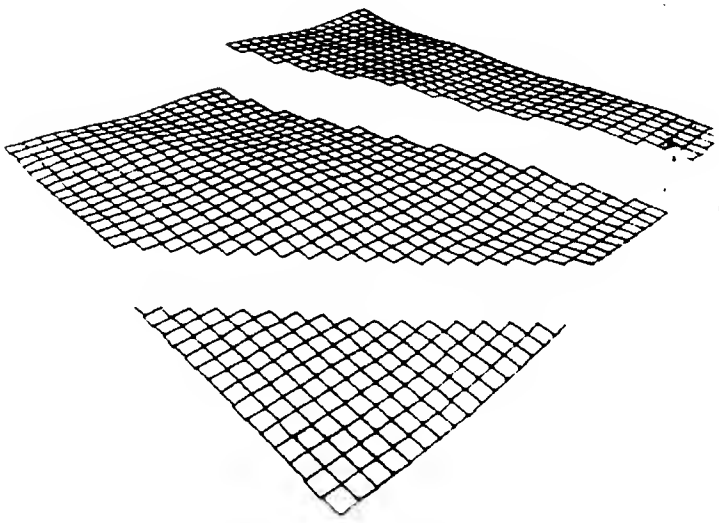
Notice here that it doesn't matter if a neighboring processor is faulty or if the wire connecting them is broken. For example, in Figure 4-7, processor A



a) exact solution



b) after 1 smoothing step



c) after 20 smoothing steps

Figure 4-6. Handling noise.

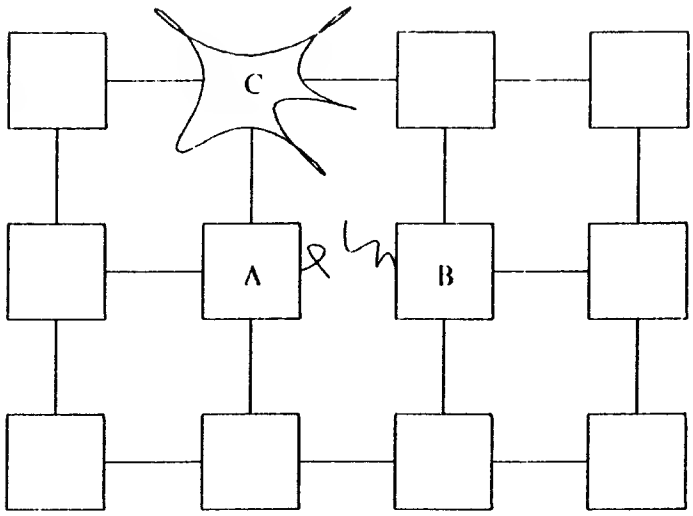


Figure 4-7. Faulty processor array.

marks B as being broken since A gets no response over their connecting wire. B's other neighbors however think that B is in fine shape and can do a good job. Of course, B thinks that it is A that is broken since A never responds to his queries. An example of a solution of a problem given 10% random broken processors out of an array of 32 by 32 is given in Figure 4-8. Notice that the overall solution stays very much correct, we only lose a small amount of resolution at the location of bad processor sites.

4.6. Concurrent Multigrid Speedup - The Pyramid Network

Like all local iterative algorithms, the coupled depth/slope method takes a long time to converge. For realistic images, it literally takes thousands of iterations to calculate a surface. In this section, we propose a new concurrent multigrid algorithm to speedup the coupled depth/slope calculations.

Terzopoulos [1984] used multigrid methods to improve the convergence rate of his surface reconstruction algorithm. This was done for normal sequential computers. Multigrid methods [Brandt 1977; Hackbusch and Trottenberg 1981]

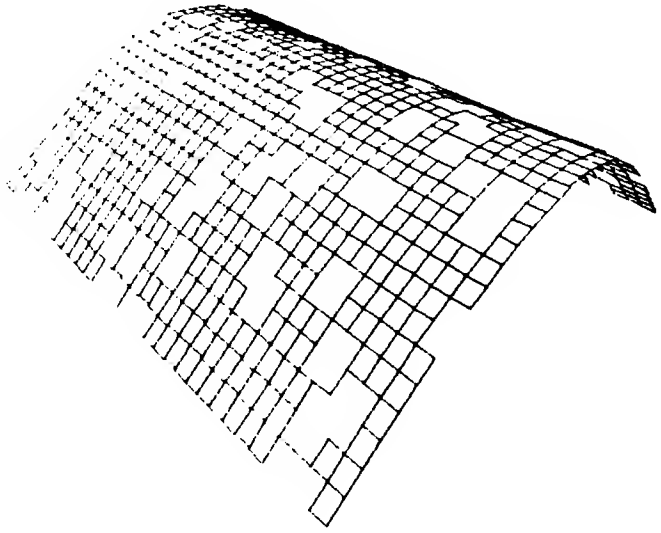


Figure 4-8. Example of fault tolerance result

are a standard set of mathematical techniques designed to speed the convergence of an iterative algorithm. The basic idea is to solve the problem first on a coarse level and then use that solution as a first approximation for a finer level.

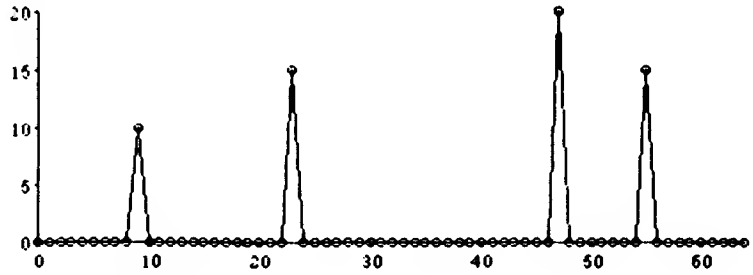
Multigrid methods are often many orders of magnitude faster than their single level counterparts. For instance, suppose we are given an N by N grid on which to calculate the solution of a surface. The computation can first be mapped to a coarser grid of $N/2$ by $N/2$ mesh points. There are two reasons for the multigrid speedup. A single iteration on this coarse grid takes one-fourth the number of the operations of a single iteration on the finer grid. In addition, since signals do not have to propagate as far on the coarse grid, it takes fewer iterations to converge to a solution. So convergence on the coarse grid is much faster since fewer iterations are needed and less computation is required for each iteration.

When a multigrid algorithm is run on a problem which has already been mapped to a parallel machine, the relative speedup is not as great as a multigrid algorithm running on a sequential machine. This is because, on a parallel ma-

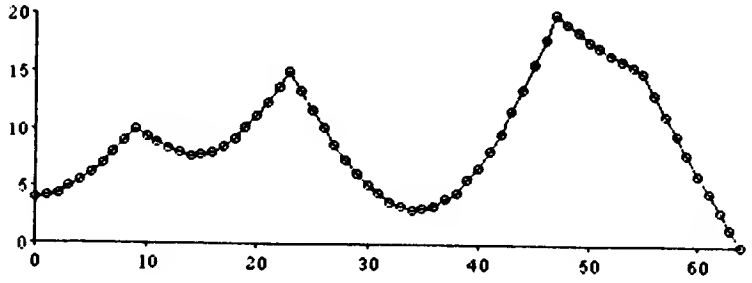
chine, the time it takes to do a single iteration on any level is the same no matter which level of coarseness is being solved. In the sequential case, iterations on the next coarser level take one-quarter the speed of the finer level, but in a parallel machine, all the nodal computations are done in parallel so all grids do iterations equally fast. However, multigrid methods still give some algorithmic speedup on a parallel machine because it takes fewer iterations to converge on coarser levels. It's just that the overall relative speedup for introducing multigrid methods is not as great as for a sequential machine.

There is a problem with mapping these multigrid algorithms to a parallel computer like the Connection Machine. The multigrid methods presently used can only process data on one level at a time. In this sense these algorithms are inherently sequential. Recently, Terzopoulos [1985] developed an algorithm which allows all of the levels to run in parallel. He does away with the complex coarse-to-fine and fine-to-coarse transfers of data normally seen in multigrid algorithms. However, Terzopoulos' algorithm requires the use of global switching parameters which control the information flow between coarse and fine levels. During the initial stages of the algorithm, a larger percentage of information will flow from the coarser levels down to the finer levels. As the algorithm progresses, more information flows up from the finer levels, correcting the coarser levels.

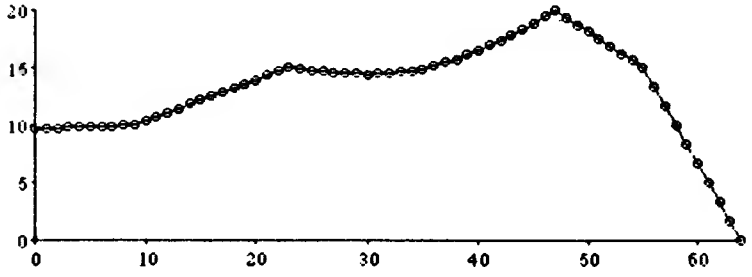
We propose a better solution. We model the multigrid problem as a network of resistors and capacitors arranged in a pyramid. This pyramid network is composed of meshes of resistors and capacitors, where different levels of meshes correspond to varying degrees of coarseness. We develop this new approach by studying the simple 1D membrane problem first. In a membrane smoothing algorithm, straight lines are fit to sparse depth constraints through a parallel iterative averaging process. Figure 4-9a shows sparse constraints along a line with 65 mesh points. If an initial guess of zeroes is assumed, the result after 50



a) sparse constraints



b) after 50 iterations



c) after 250 iterations

Figure 4-9. Single grid example

iterations, Figure 4-9b, is far from convergence. After 250 iterations, Figure 4-9c, the solution has still not converged. Convergence is reached after more than 500 iterations. The slow convergence problem is solved with the pyramid network.

For the 1D membrane problem, we can model the computation as a distributed RC line as shown in Figure 4-10. The exact values of R and C must be carefully chosen to accurately reflect the amount of information that can pass between processors during each cycle. Each iterative step corresponds to some

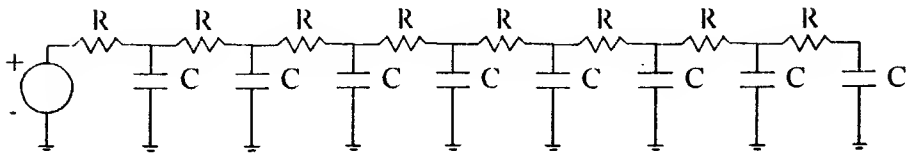


Figure 4-10. Simple RC line model of computation

small time step for the analog network. The effect of a single constraint will be modeled here as a voltage step applied to one end of the line. The initial guess is represented as the initial voltages on all of the capacitors.

Electrically, the line will exponentially converge to a solution with some RC time constant. Glasser and Dobberpuhl [1985] give an upper bound on the time constant as

$$\tau = RC\frac{N(N+1)}{2}$$

with N being the number of nodes in the mesh. The speed of the delay line varies quadratically with the its length. An obvious speed up is to add a coarser RC line such as that shown in Figure 4-11. This line will solve the problem four times as fast since it has half the length and the same R and C values as the finer level. The constraint on the finer level is used as a constraint on the coarser level at the same spatial location.

Problems occur when constraints don't occur on mesh points, This difficulty is resolved through careful selection of inter-grid information flow. Suppose we are given constraint values at locations 0 and 7 in the nine node arrangement shown in Figure 4-12. We use a hierarchy of four grids of varying levels of resolution to solve the membrane problem here. We observe that in this simple case, wires connecting different levels need only be unidirectional. The direction of information flow is determined by the location of the constraints. Information

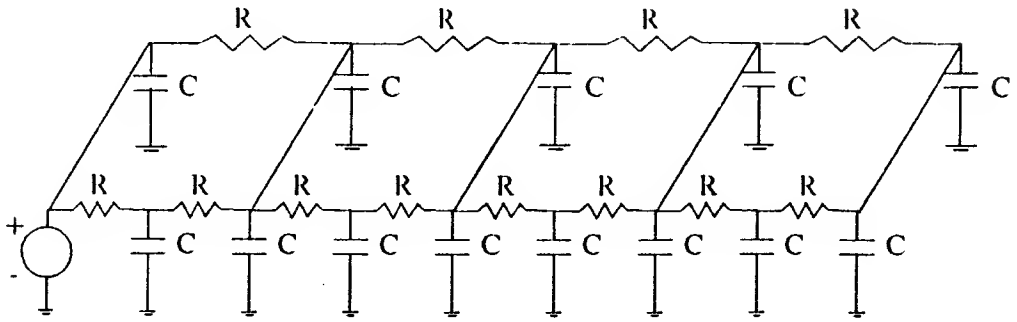


Figure 4-11. Two level speedup

flows up towards coarser grids where there are constraints and it flows down where there are no constraints. This information flow between levels is an injection process. The recipient of this information just takes these values and does not do any neighborhood averaging.

This method of transferring constraints to coarser levels is very effective. Initially, only approximate constraint values are passed up to the coarser grids. As the problem is solved on coarser grids information flows down to the finer grids, creating better constraint values to pass to the coarser grids. Terzopoulos used a static local weighting technique to pass constraint values to coarser grids. His method has the disadvantage that the approximate constraint values are not improved over time and so the coarser grids can never have an accurate representation of the surface.

How many coarse grids should be used? This pyramid algorithm uses as many levels as it needs and uses them all in parallel. Notice in Figure 4-12 that the two coarsest levels, although built in, are not used. Data only flows into them - no lower levels get information from them. If this surface patch was adjacent to a sparser region, these coarser levels would then be able to efficiently pass data there. Generally, the sparser the data in a particular region, the more

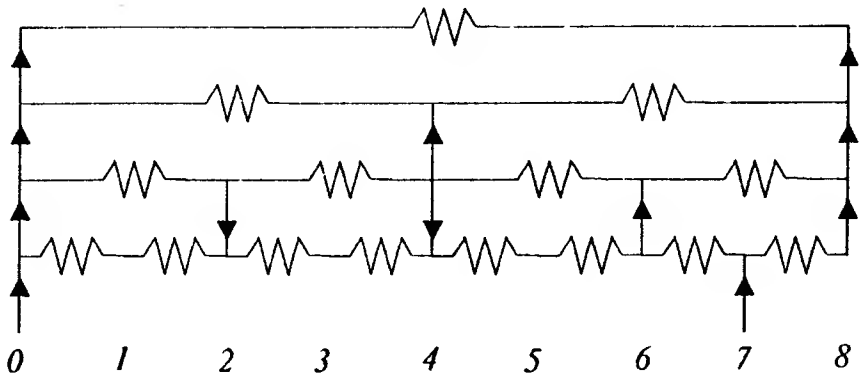


Figure 4-12. Inter-grid information flow.

levels of the pyramid that are used. If data is very dense, multigrid methods will not enable any appreciable speedup.

This pyramid network algorithm has been implemented for the 1D membrane case. The solution, shown in Figure 4-13, took only 50 iterations. This is more than an order of magnitude speed increase over the single grid solution. The pyramid algorithm converges even faster with the addition of a heuristic that uses the pyramid structure to develop a good initial guess.

We have developed a concurrent multigrid acceleration technique to speed up the surface reconstruction problem. This pyramid network is better suited for a parallel machine than conventional multigrid techniques because:

1. The pyramid network runs all of the grids concurrently.
2. The algorithm is much simpler to program than standard multigrid algorithms.

The pyramid algorithm has the following advantages over Terzopoulos' concurrent multigrid approach:

1. The system does not rely on any dynamically changing global parameters.

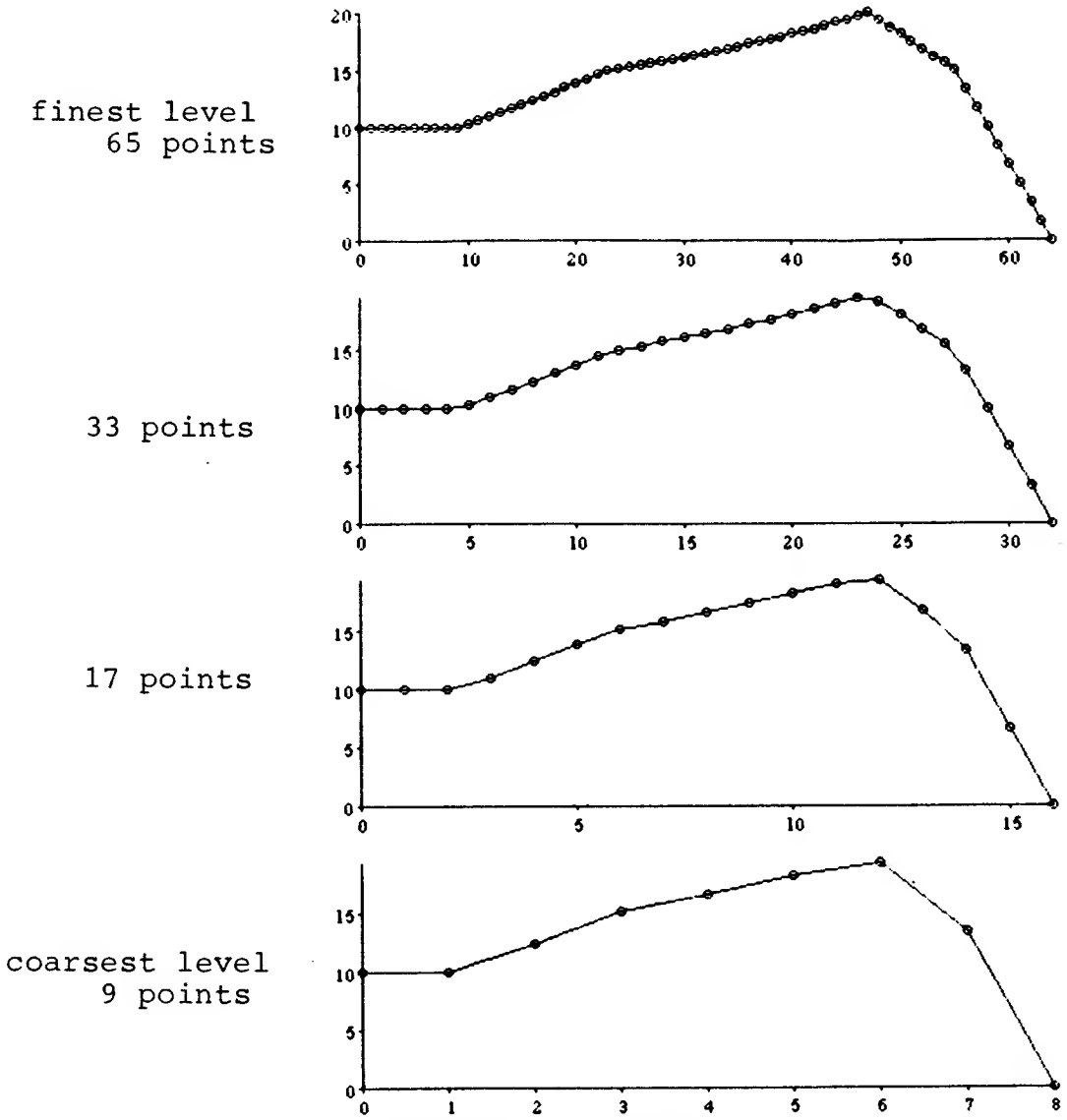


Figure 4-13. Multigrid example, convergence in 50 iterations.

2. The coarser level solutions have the potential to be more accurate since the coarse constraint values improve gradually when the finer grid solution improves.

3. An analog network implementation seems feasible.

4. The system automatically chooses how many coarse levels are used. Regions where information is sparse tend to use the most grid levels.

4.7. Digital Alternatives

Historically, direct methods for solving extremely large partial differential equation problems have been prohibitively expensive. The advent of locally connected, massively parallel machines has provided iterative algorithms with even more advantage over direct methods. Recently, Pan and Reif [1985] have proposed the parallel nested dissection algorithm for solving linear systems. Their algorithm for solving large sparse systems maps particularly well on a computer such as the Connection Machine. Such a technique could be applied to solve the system of coupled difference equations given by the coupled depth/slope method.

In general, direct methods are much more difficult to implement and require more memory than iterative methods. Iterative methods can easily make good use of approximate solutions. This is important in solving problems like optical flow. The optical flow in one time frame should be very close to the optical flow in the next frame. Direct methods cannot make use of good approximations. For extremely large problems, iterative methods are more efficient, that is, less work is required to achieve a solution of a given accuracy. Low accuracy solutions, can be generated very rapidly. The iterative multigrid methods have optimal time complexity rates for the solution of linear systems. These multigrid solvers can also work with the same efficiency on nonlinear and eigenvalue problems. Multigrid algorithms have been successfully adapted to nonlinear flow problems such as transonic flows and incompressible Navier-Stokes equations [Brandt 81].

Terrence Boult describes yet another way to perform surface reconstruction. He uses continuous thin plate splines [Boult 1985a], but does not attempt to integrate orientation information, does not provide for discontinuities of different types and makes no provisions for handling noisy measurements. Therefore,

from the definition given in Chapter 1, he does not address the full surface reconstruction problem, only surface interpolation.

Boult compares the gradient projection based algorithm used by Grimson with his own method of reproducing kernels [Boult 1985b]. He argues that the reproducing kernel method has better parallel time complexity than the single level algorithm of Grimson. Assume k depth points are given in an $m \times m$ mesh. Define the number of grid points m^2 to be n and assume a typical 10% sampling of depth points so $k = n/10$. The time consuming computation for this method is the solution of a $(k-3)$ by $(k-3)$ dense linear system of equations which takes $O(k^2)$ or equivalently $O(n^2)$ time to solve (assuming zero communication time). This compares favorably with $O(n \log n)$, the parallel time complexity that the Boult gives for Grimson's algorithm. The multigrid methods used by Terzopoulos lowers this time complexity to $O(n)$ on a sequential machine [Brandt 1977].

In this analysis, communication time complexity is ignored, despite the fact that this seems to be the major bottleneck in Boult's approach. The thin plate spline generating algorithm does not exploit the locality that exists in the problem. This contrasts with the coupled depth/slope algorithm. Surface reconstruction using iterative multigrid methods has better computation and communication time complexities.

Recently Jim Clark has developed a novel approach to surface interpolation [Clark 1985]. His idea consists of three steps. First, he maps the sparse sensory data to regularly spaced points in the 2D grid. Second, he uses standard techniques from sampling theorems in digital signal processing and does a single convolution of regularly spaced points to recreate a surface. Finally, an inverse mapping is performed to map the reconstructed surface back to the original domain. Unfortunately, his work does not extend to handle arbitrary

discontinuities. Furthermore, his method fails to extrapolate the unconstrained edges of the surface as, for example, the biharmonic operator would.

4.8. Summary

This chapter has developed a digital algorithm based upon the coupled slope depth model to solve the surface reconstruction problem. The algorithm is extremely well suited for a locally-connected massively-parallel computer such as the Connection Machine. Since this algorithm comes directly from the coupled depth/slope model, the algorithm can handle constraints and discontinuities at any order of derivative of the surface. In addition, the digital coupled depth/slope algorithm was shown to have the following advantages for a Connection Machine type implementation:

1. It requires *only* nearest neighbor communication between processors. No information from processors other than the four nearest neighbors is ever needed. This includes depth and orientation constraints, discontinuity locations as well as current parameter values.

2. The algorithm was shown to be extremely fault tolerant. Broken processors merely cause a loss of resolution; the solution is still roughly correct.

3. Each node requires only fixed point additions, subtractions and averages. This constraint is extremely important for a computer such as the Connection Machine which uses simple one-bit processors.

CHAPTER 5

ANALOG IMPLEMENTATION

The coupled depth/slope algorithm discussed in Chapter 4 runs much faster when implemented with analog circuitry. Instead of simulating an analog network with a digital computer, this chapter describes possible analog implementations of the coupled depth/slope network. First, we discuss how to build the necessary subtractor constraint device and then we discuss how to set the constraint values.

5.1. Building the Subtractor Constraint Element

The hardest part of building the coupled depth/slope network is the construction of the subtractor elements which are responsible for the coupling between the slope and depth planes. As Chapter 3 states, any of the terminals of this device can be inputs or outputs. One way of building this device is with an ideal transformer in the configuration shown in Figure 5-1. Here we rely on the mutual inductance of the two coils to produce the desired coupling effect [Senturia and Wedlock 1975, p. 135]. With a *turns ratio* of one, the voltages across the two inductors will be the same, so $A - B = C$. The transformer circuit requires alternating current to function and dissipates no power.

The use of transformers makes the coupled network equivalent to the elastic plate analogy built by MacNeal [1951] in the early 1950's. His elastic plate analogy network contained twelve nodes and could be extended to solve problems such as beam deflections and transient vibrations. There are several problems with any transformer-based network. The most severe problem is building error-free transformers. A second problem is that it is virtually impossible to build

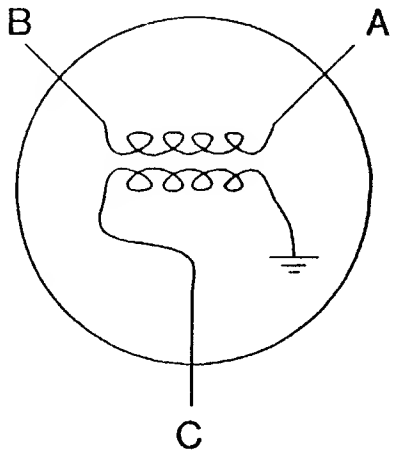


Figure 5-1. Transformer implementation of subtractor

a reasonable transformer on a VLSI chip. This precludes the possibility of constructing a large scale system.

Another way of constructing the ideal subtractor element is with the help of negative resistors as shown in Figure 5-2. By Kirchoff's current law the sum of the currents into the central node node X is zero, so

$$\frac{X-A}{-R}+\frac{X-B}{R}+\frac{X-C}{R}+\frac{X}{-R}=0$$

or rearranging terms,

$$A-B=C$$

There are a number of ways of building negative resistances. One possibility is to use a careful combination of capacitors and inductors as shown in Figure 5-3. From above, we require:

$$Z_L=-Z_C$$

or

$$j\omega L=-\frac{1}{j\omega C}$$

$$\omega=\sqrt{\frac{1}{LC}}$$

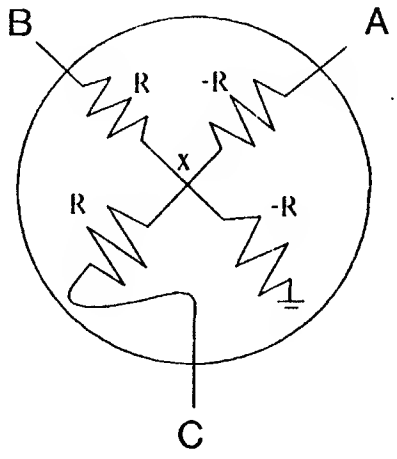


Figure 5-2. Negative resistance implementation.

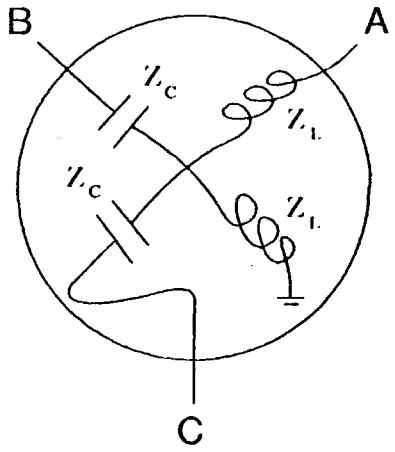


Figure 5-3. Frequency domain implementation.

Not only is alternating current required, but the frequency has to exactly match the values of L and C. This is a difficult circuit to build correctly, and again we cannot build it onto an integrated circuit.

Another implementation strategy is to implement the device with two ana-

log subtractors and one analog adder, as was shown in Figure 3-2. An adder is needed to compute $B + C$ and the two subtractors are used to compute $A - C$ and $A - B$. Adders and subtractors are simple to build with operational amplifiers. The full subtract constraint element circuit is shown in Figure 5-4. A working version of this circuit has been built and tested.

Problems occur in real life when we try to construct ideal devices out of non-ideal parts. Each op amp adder and subtractor circuit has tiny errors in resistance values, finite gain, finite input impedance and non-zero output impedance. These errors cause slight inaccuracies in the calculated values and the constraint $A - B = C$ will not be true in general. The small error doesn't bother us from an algorithmic standpoint since the inaccuracies are very small and are averaged together in each constraint element. There is a problem from the electrical point of view, though. The circuit described sets the output of all of its op amps to values which are slightly different from their normal output. Since op amps have an extremely small output impedance a lot of current will flow, possibly burning the op amp. The problem is reduced by explicitly increasing the op amp output impedance with an extra resistor in the feedback path.

Perhaps a better approach is to implement directly the negative resistance values given in Figure 5-3 with negative impedance converters [Siebert 1986]. With two negative impedance circuits, the subtract constraint element can be constructed with two op amps and eight resistors as shown in Figure 5-5. Also, the rivalry and error problems described for the circuit in Figure 5-4 aren't present in this implementation.

The big advantage of these op amp circuits is that operational amplifiers are readily fabricated with VLSI technology. The idea of surface reconstruction on a chip is not too far in the future. The op amps need not consume excessive

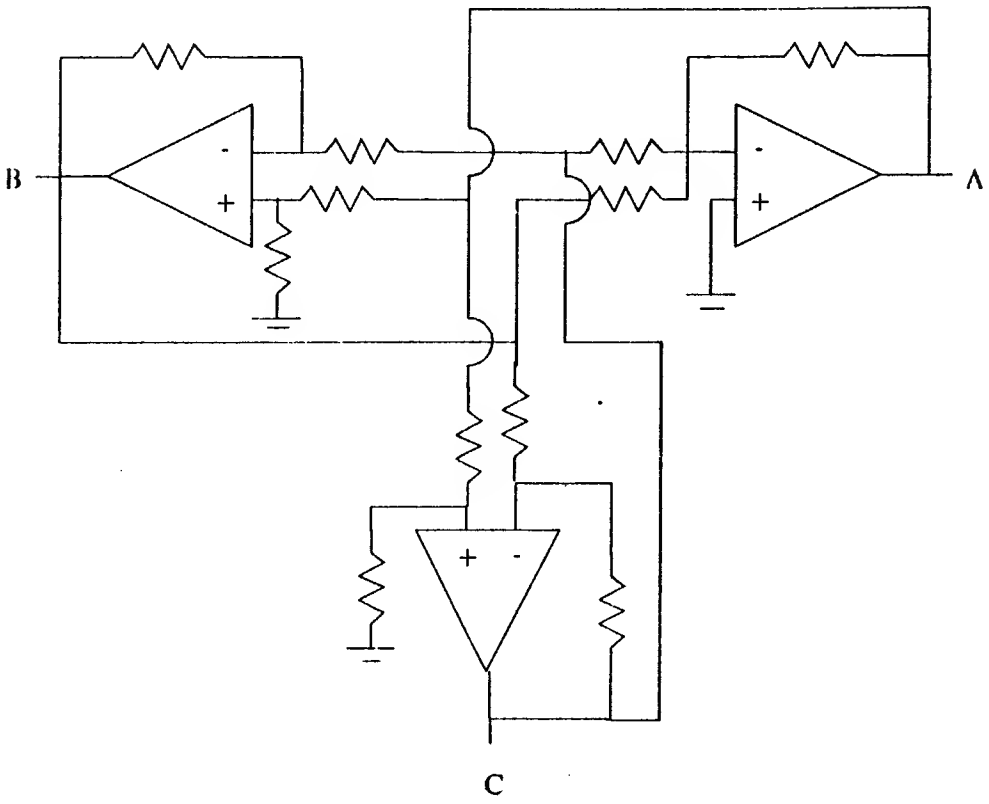


Figure 5-4. Operational amplifier implementation.

chip space since the the DC biasing circuitry can be shared among many op amps.

5.2. Setting Constraint Values

In Chapter 3, we proposed to set constraint values with a voltage source connected through a resistor to the constraint mesh point, the Thèvenin equivalent, see Figure 5-6a. The depth measurement $\hat{u}$ corresponds to the value of the voltage source and the conductance of the resistor is α, with α defined as

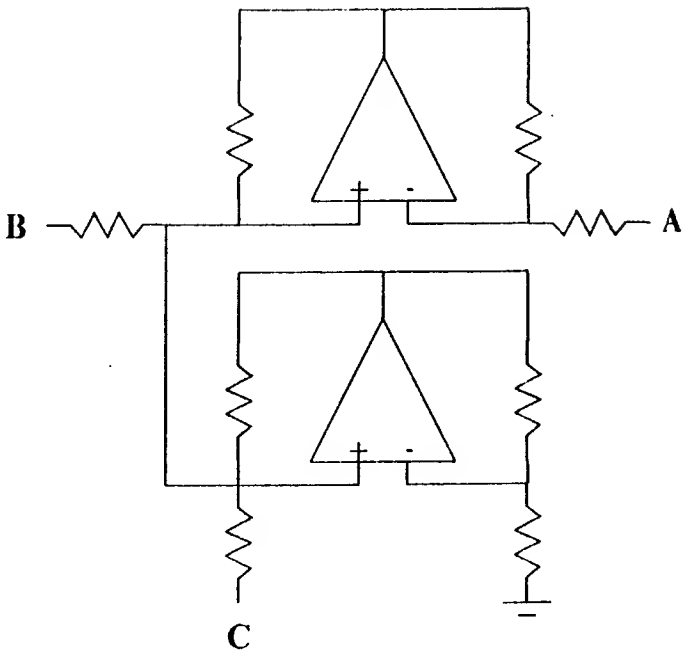


Figure 5-5. Negative impedance converter implementation.

a measure of the confidence in the measurements. The more typical connection proposed in surface approximation networks [Marroquin 1985; Koch, Marroquin and Yuille 1986; Poggio and Koch 1985; Hutchinson 1986] is to use the Norton equivalent circuit, shown in Figure 5-6b.

The Thèvenin equivalent used to set constraints has several important implementation advantages.

1. While resistor meshes are generally insensitive to resistor value variations [Karplus 1958], the Norton circuit is very sensitive to the resistance variations that are common over VLSI process corners [Hutchinson 1986]. The voltage seen at u varies directly with variations in resistor values. The similar problem does not occur for the Thèvenin circuit. Variation in a resistor value simply changes the confidence in $\hat{u}$.

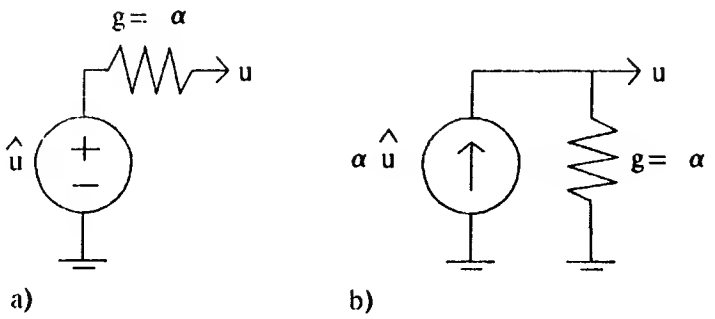


Figure 5-6. a) Thèvenin equivalent b) Norton Equivalent

2. Using the Thèvenin equivalent circuit also has the advantage of reduced power consumption. We want to maximize the voltage swing in the resistor mesh to achieve the largest resolution possible. Power dissipation is a problem, therefore, since power is $\sum(V^2/R)$. We set the resistor values in the mesh as large as possible to minimize circuit power. It can be shown that the Norton circuit dissipates much more power than the Thèvenin circuit when driving a large resistive load.

3. Finally, it is easier to implement the Thèvenin circuit since the variable α parameter sets only the resistance value and is not involved with the value of the voltage source. The α value plays a part in both the current source and the resistance values for the Norton equivalent circuit.

Unfortunately, it is simpler to build current sources than voltage sources with today's silicon technology. Nonetheless, the Thèvenin circuit has the potential for more process variation immunity, less power dissipation, and simpler overall system implementation.

5.3. Alternative Analog Networks

There are other possible analog networks which implement biharmonic smooth-

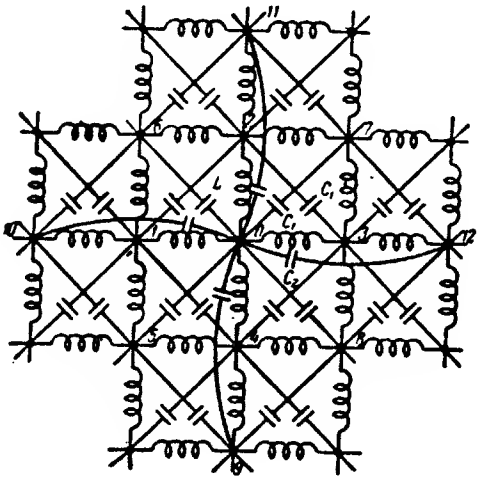


Figure 5-7. Electrical network realization in the frequency domain.

ness. The biharmonic mask can be synthesized electrically by using the tricks discussed in the last section for manufacturing negative resistance components. For example, the biharmonic mask can be realized in the frequency domain with a network of resistors, capacitors and inductors [Volynskii 65]. Figure 5-7 shows a typical circuit. Unfortunately, these networks are hard to build in today's VLSI technology because inductors are difficult to fabricate. Also, small errors in capacitance and inductance values will cause large variations in the solution. A more plausible network would take advantage of the negative impedance converters used in Figure 5-5.

The dual-Poisson network described in Chapter 3 can be implemented straightforwardly. However, depth values (or slope values) must be given at each point around the entire boundary. This network cannot implement the biharmonic free-edge condition required for surface reconstruction. Also, there can be no discontinuities in the interior of the image unless the slope or depth is known at every point on both sides of all discontinuities. The dual Poisson network may be appropriate for restricted domains but is not useful for general

surface reconstruction applications.

5.4. Summary

We have proposed implementation strategies for building an analog version of the coupled depth/slope method. The rough estimated speed of such a network is on the order of microseconds. Unfortunately, this network will be plagued by the problems common to all analog networks: I/O boundedness, inflexibility and limited resolution. Clearly, the speed of the system will be determined by how fast data can be loaded and unloaded into the network. It is very difficult to modify the algorithms when they are compiled into hardware. Noise limits the precision of voltages stored on nodes in the system. Because of these problems, analog networks have been little more than a curiosity in the past.

CHAPTER 6
FUTURE DIRECTIONS

The coupled depth/slope model provides a new way to think about surface reconstruction. As always, new solutions tend to open new areas of research. Previously, we ignored the black boxes that performed discontinuity detection, "shape from", and stereo. We assumed that the results of these early vision procedures were available as input to the surface reconstruction module. An area for future research is to combine these early vision processes with the surface reconstruction module to improve the performance of the overall system.

6.1. Combining Vision Modules

Discontinuity detection is an active area of research in computer vision today. Terzopoulos [1985] and Marroquin [1985] both looked at combining discontinuity detection with surface reconstruction. Terzopoulos developed a heuristic for recognizing discontinuities. After a surface has been fully reconstructed, he looks for significant changes in curvature in order to establish the placement of discontinuities. Marroquin used binary line processes to represent surface discontinuities. He minimizes a non-quadratic cost function to locate discontinuities.

Either of the above methods can be incorporated into the coupled depth/slope model. Terzopoulos' method can be implemented by building the resistors in the resistor mesh as fuses. The resistors open circuit when the current passing through them exceeds some threshold. This threshold should be set to some finite value only at the locations of edges in the primal sketch. This will decrease

the possibility of false discontinuity detections. In this way, discontinuities at any level of smoothness could be detected. With the concurrent multigrid approach, discontinuities at different levels of resolution can be detected and represented.

Similarly, the stereo module could be effectively combined with the coupled depth/slope algorithm. Stereo modules typically use a hierarchical style approach to solve the stereo matching problem. The simpler, more obvious matches are done on coarser grids. These coarser matches are then carried through onto the finer grids. If the surface reconstruction module were integrated and performed concurrently with the stereo module, the following advantages would be seen. First, coarser surface reconstruction could start as soon as the stereo module registered some coarse matches with disparities. This would speed the running time of the surface reconstruction module. Second, coarse reconstructed surfaces could be fed back to the stereo module to aid matching at finer levels. This will help the stereo module to find more depth points with fewer errors, resulting in a faster convergence and a more accurate reconstructed surface.

6.2. Other Applications

The coupled depth/slope network is not restricted to solving only the surface reconstruction problem. This network generalizes to arbitrary dimensions and arbitrary levels of smoothness. For example, we have seen that Laplacian, Poisson, and biharmonic type problems are easily solved with this network. These partial differential equations arise in such computationally demanding fields as hydrodynamics, aerodynamics, weather forecasting and structural analysis. The hardware described in Chapter 5 is being studied for use in thin beams or cantilever problems in structural analysis. A simple example is shown in Figure 6-1 where an elastic beam is fixed on its right and left edges and a downward force

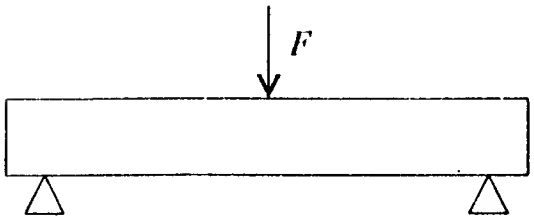


Figure 6-1. Elastic Beam Problem

is applied at the center of the beam. The final displacement of the beam must be found.

The coupled depth/slope network easily handles problems such as these. The input forces are fed as input at the bottom of the network (see Figure 6-2) using appropriately valued voltage sources. There is no need for any of the resistor meshes because these are dense constraints. No minimization is necessary; only integration is performed. Extra constraints are necessary to regain the constants of integration. Alternatively, we could feed dense deflection data into the top of the array and read the forces from the bottom. Or we could feed only sparse data and add a resistor mesh for some appropriate level of smoothness. In summary, we can give the network any data that we have about the moments, forces, slopes, or deflections of the beam. These constraint values, with their corresponding confidence values, are incorporated into the final beam solution. We can directly read dense solutions of the loading, the shearing force, the bending moment and the beam deflection.

6.3. Problems to be addressed

The coupled depth/slope network has raised a number of questions which remain to be answered.

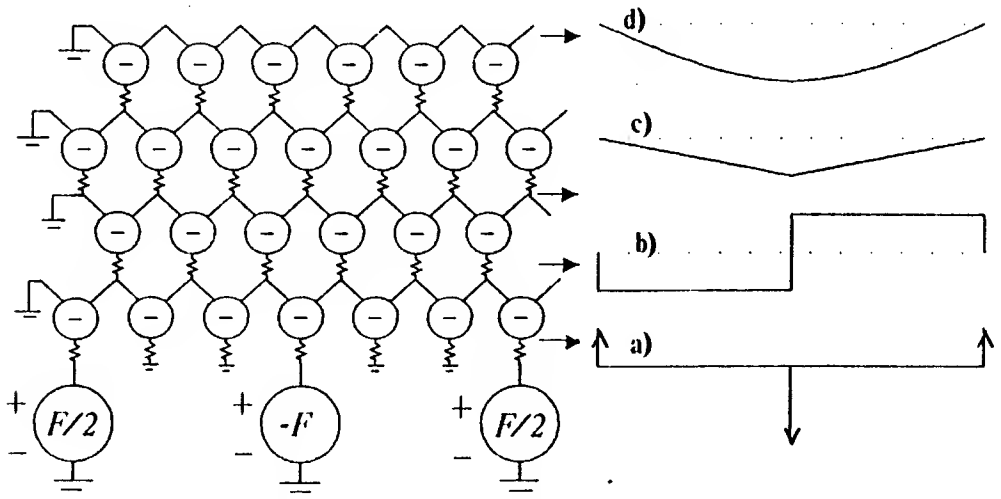


Figure 6-2. Elastic Beam Solution: a) Loading diagram b) Shearing force diagram c) Bending moment d) Deflection line

- How similar is the post-smoothing approach (Section 4.4) to the conventional least squares averaging approach?

- This report has concentrated on parallel implementations of the coupled depth/slope network. How well does this algorithm work with a Gauss Seidel iterative scheme on a sequential scheme? Does the method have a low enough smoothing rate to be appropriate for a standard multigrid implementation?

- The concurrent multigrid algorithm described in Section 4.6 has been implemented for solving one-dimensional membrane problems. It must be modified for the 2D membrane case and for the coupled depth/slope algorithm. What are the exact speeds of these algorithm on the Connection Machine and on a pyramid machine? How do these rates compare to the speedup afforded by conventional multigrid algorithms?

- The 2D form of the generalized coupled depth/slope network must be de-

veloped. It should be possible to show convergence for both the analog network and its digital implementation.

- What is the best way to build the analog subtract constraint device? Will the circuits proposed in Chapter 5 converge to a solution or oscillate and diverge?

CHAPTER 7
CONCLUSION

In summary, we have developed the coupled depth/slope model for surface reconstruction. This is a computational model with a number of advantages over earlier models. If only depths are given, the model reconstructs a surface with the same biharmonic smoothness used by Terzopoulos and Grimson. In addition, as depths are computed, orientation constraints can be incorporated with no extra work. Discontinuities in depth and orientation can be used to compute the reconstructed surface. Furthermore, the coupled depth/slope model generalizes to handle constraints and discontinuities of any order derivative.

The model appears to be biologically feasible in that it fits the four criterion given in Chapter 1: parallelism, uniformity, locality and fault tolerance. Its biharmonic smoothness properties are observed in psychophysical tests of humans. Also, Marr argues that the human $2\frac{1}{2}$-D sketch redundantly represents both slopes and depths. This compares favorably with the dense depth and slope representations generated by the coupled depth/slope network.

Besides appearing to be biologically feasible, the coupled depth/slope model leads naturally to compuational solutions on both digital computers and in analog circuitry. In fact, a digital algorithm based upon the coupled depth/slope model was implemented and was found to be extremely well suited for a locally-connected massively-parallel computer such as the Connection Machine. Since this algorithm comes directly from the coupled model, the algorithm can handle constraints and discontinuities at any order of derivative.

The coupled depth/slope algorithm was shown to have the following advantages for a Connection Machine implementation. It required *only* nearest

neighbor communication between processors. No information from processors other than the four nearest neighbors is ever needed. This includes depth and orientation constraints, and discontinuity locations as well as current depth and slope values. The algorithm was shown to exhibit a high degree of fault tolerance. Broken processors cause only a loss of resolution; the overall solution is still correct. Each node requires only fixed point additions, subtractions and averages. This constraint is extremely important for a computer such as the Connection Machine which uses simple 1-bit processors.

In addition to the above results, we have suggested a new concurrent multigrid strategy for speeding up a massively parallel digital implementation. The pyramid network allows iterations to be performed on every level simultaneously. The implementation is much simpler than that of the conventional multigrid algorithm. The system converges to a solution without the addition of any dynamically changing parameters. The coarser level solutions are potentially more accurate since the coarser constraint values improve gradually when the finer grid solution improves. An analog implementation of the pyramid network is plausible.

In terms of biological vision systems, our computational model for surface reconstruction possesses remarkable similarities to the human vision system. It remains to be seen if biological systems actually do create a dense surface description or whether they perform some more complicated process.

In terms of the construction of artificial machine vision systems, it is quite clear that we will be able to build systems which can perform surface reconstruction at video scan rates. Unfortunately, no artificial later vision algorithms exist that can make good use of this immense volume of data at these fast rates. In effect, we have shifted the vision bottleneck from early to immediate vision and we are now pushing it further towards later vision.

References

Boult, T. E. and Kender, J., [1985a], "On Surface Reconstruction Using Sparse Depth Data " in *DARPA Image Understanding Workshop Proceedings*, Miami, FL, 197-208.

Boult, T. E., [1985b], "Visual Surface Interpolation: A comparison of Two Methods" in *DARPA Image Understanding Workshop Proceedings*, Miami, FL, 466-478.

Brandt, A., [1977], "Multi-level Adaptive Solutions to Boundary-value Problems", *Math. Comp.*, **31**, 333-390.

Brandt, A., [1981], "Multigrid Solvers on Parallel Computers", in *Elliptic Problem Solvers*, ed. M. H. Schultz, pp. 39-83, Academic Press, Inc., New York, NY.

Canny, J. F., [1983], "Finding Edges and Lines in Images" Artificial Intelligence Lab, no. TR-720, MIT, Cambridge, MA.

Clark, J.J. [1985], "Multi-Resolution Stereo Vision with Application to the Automated Measurement of Logs", Ph.D. Thesis, MIT Cambridge, MA.

Courant, R. and Hilbert, D. [1953], *Methods of Mathematical Physics*, Vol. 1, Interscience Publishers, Inc., New York.

Courant, R. and Hilbert, D. [1962], *Methods of Mathematical Physics*, Vol. 2, Interscience Publishers, Inc., New York.

Glasser, L. and Dobberpuhl, D. [1985], *The Design and Analysis of VLSI Circuits*, Addison-Wesley.

Goos, G. and Hatmanis, J. [1976], *Matrix Eigen System Routines - EISPACK Guide*, Springer-Verlag.

Grimson, W.E.L., [1981], *From Images to Surfaces: A Computational Study of the Human Early Visual System.* MIT Press. Cambridge, MA.

Hackbusch, W. and Trottenberg, U. [1981] *Multigrid Methods*, Springer-Verlag.

Hildreth E., [1980], "Implementation of a Theory of Edge Detection", MIT Artificial Intelligence Laboratory TR-579.

Hillis, W.D., [1985], "The Connection Machine", Ph.D. Thesis, MIT, Cambridge, MA.

Horn, B.K.P., [1974], "Determing Lightness from an Image", *Computer Graphics and Image Processing*, **3**.

Horn, B.K.P., [1975], "Obtaining Shape from Shading Information," in *The Psychology of Computer Vision*, P.H. Winston (Ed.), McGraw-Hill.

Hutchinson, J., [1986] "Early Vision Problem Solving with Analog Networks." M.S. Thesis, MIT, Cambridge, MA.

Ikeuchi, K., [1983], "Constructing a Depth Map from Images" MIT AI Lab Memo No. 744.

Knight, T., [1983], "Design of an Integrated Optical Sensor with On-Chip Preprocessing", PhD Thesis, MIT, Cambridge, MA.

Koch, C., Marroquin, J., and Yuille, A. [1986], "Analog Neuronal Networks in Early Vision", *Proc. Natl. Acad. Sci.* USA **83**, to appear.

Kuo, C., [1985], "Parallel Algorithms and Architectures for Solving Elliptic Partial Differential Equations", M.S. Thesis, MIT, Cambridge, MA.

Marr, D., [1982], *Vision*, W.H. Freeman and Company, San Francisco.

MacNeal, R.H., [1951], "The Solution of Elastic Plate Problems by Electrical Analogies." *J. Applied Mechanics,* **18**. 59-67.

Marroquin, J., [1985], "Probabilistic Solution of Inverse Problems" MIT AI Lab Technical Report 860.

Pan, V., and Reif, J., [1985], "Efficient Parallel Solution of Linear Systems", *Proceedings of the 1985 Symposium of the Theory of Computers.*

Poggio, T. [1984], "Vision by Man and Machine." MIT AI Lab Memo 776.

Poggio, T., Torre V., and Koch, C. [1985], "Computational Vision and Regularization Theory", *Nature*, **317**, 314-119.

Poggio, T. and Torre, V., [1984], "Ill-posed Problems and Regularization Analysis in Early Vision." MIT AI Lab Memo 773.

Poggio, T. and Koch, C., [1985], "Ill-posed problems in early vision: from computational theory to analogue networks," *Proc. R. Soc. Lond.* B **226**, 303-323.

Senturia, S. and Wedlock, B., [1975], *Electronic Circuits and Applications*, John Wiley & Sons, Inc., New York.

Siebert, W. [1986], *Circuits, Signals, and Systems*, MIT Press, Cambridge, MA.

Smith, G. D., [1978], *Numerical Solutions of Partial Differential Equations: Finite Difference Methods*, Oxford University Press, New York.

Terzopoulos, D., [1983], "Multi-level Computational Processes for Visual Surface Reconstruction", *Computer Vision, Graphics, and Image Processing* **24**, 52-96.

Terzopoulos, D., [1984], "Multiresolution Computation of Visible-surface Representations", Ph.D. Thesis, MIT, Cambridge, MA.

Terzopoulos, D., [1985], "Concurrent Multilevel Relaxation", *Proc. DARPA Image Understanding Workshop.*

Terzopoulos, D., [1986], "Regularization of Inverse Visual Problems Involving Discontinuities," PAMI 8(2), May, to appear.

Tikhonov, A.N. and Arsenin, V.Y., [1977], *Sollutions of Ill-Posed Problems.* Winston & Sons, Washington, D.C.

Ullman, S. [1979], *The Interpretation of Visual Motion.* MIT Press. Cambridge, MA.

Varga, R., [1962], *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, NJ.

Volynskii, B. A., and Bukhman, V. Ye., *Analogues for Solution of Boundary-Value Problems*, 1965.

Young, D. M. [1971], *Iterative Solution of Large Linear Systems.* Academic Presss, London and New York.

This blank page was inserted to preserve pagination.

CS-TR Scanning Project
Document Control Form

Date : 10/18/95

Report # AI-TR-908

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☒ Technical Report (TR) ☐ Technical Memo (TM)
☐ Other:_____

Document Information

Number of pages: 83 (90-IMAGES)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☒ Laser Print
☐ InkJet Printer ☐ Unknown ☐ Other:_____

Check each if included with document:

☒ DOD Form (2) ☐ Funding Agent Form ☒ Cover Page
☒ Spine (ON COVER) ☐ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages (by page number): FOLLOW ABSTRACT, ACKN, CONTENTS PAGES

Photographs/Tonal Material (by page number):_____

Other (note description/page number):

Description :	Page Number:
IMAGE MAP:	(1-8)
UNN'ED TITLE, INFO, ABST, BLANK, ACK, BLK, CONTENTS, BLK PAGES.	
	(9-83)
PAGES #'ED 5-75, NO PAGES 76, 77-80	
	(84-90)
SCANCONTROL, COVER, DOD(2), TRGT'S (3)	

Scanning Agent Signoff:

Date Received: 10/18/95 Date Scanned: 10/23/95 Date Returned: 10/26/95

Scanning Agent Signature: Michael N. Cook

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 908	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): The Coupled Depth/Slope Approach to Surface Reconstruction	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): John G. Harris	8. CONTRACT OR GRANT NUMBER(s): DACA76-85-C-0010, N00014-85-K-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Advanced Research Projects Agency, 1400 Wilson Blvd., Arlington, VA 22209	12. REPORT DATE: June 1986 13. NUMBER OF PAGES: 80
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Office of Naval Research, Information Systems, Arlington, VA 22217	15. SECURITY CLASS. (of this report): UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

None

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Surface Reconstruction, Parallel Algorithms, Analog Networks

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Reconstructing a surface from sparse sensory data is a well known problem in computer vision. Early vision modules typically supply sparse depth, orientation and discontinuity information. The surface reconstruction module incorporates these sparse and possibly conflicting measurements of a surface into a consistent, dense depth map. The coupled depth/slope model developed here provides a novel computational solution to the surface reconstruction problem. This method explicitly computes (over)

(abstract cont.)

dense slope representations as well as dense depth representations. This marked change from previous surface reconstruction algorithms allows a natural integration of orientation constraints into the surface description, a feature not easily incorporated into earlier algorithms. In addition, the coupled depth/slope model generalizes to allow for varying amounts of smoothness at different locations on the surface.

This computational model helps conceptualize the problem and leads to two possible implementations - analog and digital. The model can be implemented as an electrical or biological analog network since the only computations required at each locally connected node are averages, additions and subtractions. A parallel digital algorithm can be derived by using finite difference approximations. The resulting system of coupled equations can be solved iteratively on a mesh-of-processors computer, such as the Connection Machine. Furthermore, concurrent multi-grid methods are designed to speed the convergence of this digital algorithm.

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**